R

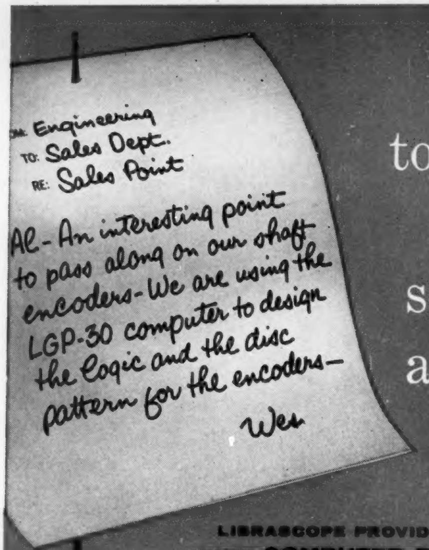
COMMUNICATIONS

Of The Association For

COMPUTING MACHINERY

Volume 1 . Number 8 August 1958

Published by the Association for Computing Machinery



Engineering
TO: Sales Dept.
RE: Sales Point

Al - An interesting point to pass along on our shaft encoders - We are using the LGP-30 computer to design the logic and the disc pattern for the encoders -

Wes

tough design problems solved with a computer

LIBRASCOPE COMPUTER DESIGNED SHAFT-TO-DIGITAL ENCODERS

LIBRASCOPE PROVIDES GREATER ACCURACY WITH COMPUTER-DESIGNED COMPONENTS

Use of the LGP-30 computer to solve encoder design problems is another Librascope advancement in the design and manufacture of shaft-position-to-digital encoders. *Computer-designed* logic and disc patterns assure accuracy, reliability, long life, and non-ambiguity in a syncro-mounted package. Analog-digital or digital-analog information may be transmitted at rates up to 1 mc and in some cases above. Wherever serial readout is preferred, Librascope *computer-designed* features and quality make these encoders the ideal choice for input devices to digital computers and data processing systems. Available in straight Binary, Binary Coded Decimal, Gray, and Sine-Cosine codes as shown in the chart below. Special units may be designed to fit your problems. *Write for catalog information.*

CODE	MODEL†	Resolution Per Input Shaft Rev. (Counts)	Accuracy Over Full Range	Dimensions Diameter x Length
BINARY	7 digit	128	1 part in 128	2" x 2⁴⁹⁄₆₄"
	13 digit	128	1 part in 8192	2" x 3¹⁹⁄₃₂"
	17 digit	128	1 part in 131,072	2" x 4¹³⁄₁₆"
	19 digit	128	1 part in 524,288	2" x 4¹³⁄₁₆"
10 BIT	10 digit	1024	1 part in 1024	4³⁄₁₆" x 1¼"
SINE-COSINE	*	*	1 count in total capacity	4³⁄₁₆" x 3⁵⁄₁₆"
BINARY CODED DECIMAL	0-2000	200	1 part in 2000	3¹⁄₁₆" x 4²⁷⁄₃₂"
	0-3600	200	1 part in 3600	3¹⁄₁₆" x 4²⁷⁄₃₂"
	0-20,000	200	1 part in 20,000	3¹⁄₁₆" x 4²⁷⁄₃₂"
	0-36,000	200	1 part in 36,000	3¹⁄₁₆" x 6⅜"
GRAY	8	256	1 part in 256	3¹⁄₁₆" x 1¹¹⁄₁₆"

†SPECIAL UNITS AVAILABLE

Precision gearing
Shaft Speed: 120 rpm continuous
Operating temp: −55°C to +75°C
Shock and Vibration: up to 15 G, 5 to 500 cps.

Life Expectancy: Function of lead current. For 13 digit unit @ 2 ma. per brush, life approx. 5 x 10⁸ breaks or makes at approx. 120 rpm.

* 8 digits per quadrant plus the limit 1 plus polarity information. One turn of the input shaft generates 4 quadrants of information. Sine and cosine functions available simultaneously and independently.

DIGITAL READOUT EQUIPMENT AVAILABLE

Librascope's use of the LGP-30 computer simplifies complex design and production problems, and assures computer-engineered quality in meeting design and delivery schedules.

LIBRASCOPE

GPE PRECISION GENERAL CORPORATION

A SUBSIDIARY OF GENERAL PRECISION EQUIPMENT CORPORATION

LIBRASCOPE, INC. • 808 WESTERN AVENUE • GLENDALE, CALIFORNIA

COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY

Communications of the Association for Computing Machinery is published monthly for the Association for Computing Machinery, 5800 N. Marvine St., Philadelphia 41, Pa. Second-Class mail privileges authorized at Philadelphia, Pennsylvania.

In addition to the *Communications*, the Association also publishes the *Journal of the Association for Computing Machinery*. The subscription price for both publications is $5.00 per year for members, of which $3.00 is for the *Journal* and $2.00 for the *Communications*. Subscription to both publications for non-members is $10.00, of which $6.00 is for the *Journal* and $4.00 for the *Communications*. Single copies of *Communications* $1.00.

Subscriptions, notices of address changes and other communications on business matters should be addressed to the Association for Computing Machinery, 2 East 63rd Street, New York 21, N. Y. Material intended for publication and communications on editorial matters should be sent in duplicate to the Editor-in-Chief, or to the Editor of the appropriate department.

By arrangement with the office of Naval Research, U. S. Navy Dept., the Digital Computer Newsletter issued quarterly by that organization is reprinted in its entirety as a supplement to the Communications.

LETTERS TO THE EDITOR

Dear Editor:

As an essential step toward finding out what additional service might be provided computer users by a technical society, the American Institute of Chemical Engineers has established an *ad hoc* committee on computer program interchange and/or publication. This committee (containing representatives of chemical companies, petroleum companies, engineering colleges, and engineering contractors) plans to report its recommendations in December, 1958, to the Council of A.I.Ch.E.

Since a large number of A.C.M. members are engaged in work which affects chemical and petroleum technology, comments are invited on (a) desirability of an additional program interchange and/or publication service, (b) standards for such a service, (c) most fruitful areas for such a service, and (d) feasibility of joint A.C.M. and A.I.Ch.E. effort.

Comments may be addressed to the writer.

Sincerely yours,

WALTER M. CARLSON
Engineering Department
du Pont Company
Wilmington 98, Delaware

Dear Editor:

Free flow of information in technical fields is essential to avoid unnecessary redundancy of effort. This is obviously true for work performed within the United States, and applies equally to work performed in other countries.

As members of the British Computer Society we would be interested in the formation of a group which would act informally to prepare extracts from articles published in the British Computer Society Journal and the ACM Journal, and the journals of any other foreign societies that may exist.

We would appreciate communications on this subject from other people similarly interested, with the expectation that the results of this work, if well distributed, will have a value several orders greater than the work expended. This letter is written, therefore, strictly as a request to others to determine:

(a) The estimated value of making such abstracts.
(b) The interest that exists within the ACM for this venture, and
(c) The support that we may expect from others in this venture.

Yours very truly,

ROGER L. BOYELL
154 N. Franklin Street
Hempstead, L. I.

B. CONWAY
56 Pine Street
New York 5, New York

TECHNIQUES DEPARTMENT

Editor's Note:

A revised summary of the material in the A.C.M. library is being printed to encourage further contributions of missing material.

As the translation of the Russian paper appearing in this section does not give any easy clues about its subject material or intent, a brief description is attempted here. It is nice to see that English-speaking peoples are not the only experts at obfuscation.

This paper deals with the production of 3-address machine language instructions (for the BESM computer) from algebraic statements of the type found in Fortran, Unicode and other languages. Assembly program characteristics are included. An algorithm is given for creating rough machine language instructions in pseudo-form and then operating upon these to alter them to the most efficient form. For at least the domain of a single formula, a check is made for duplicate strings of any length.

The most pertinent point is that the algorithm itself operates more efficiently than their previous methods. Thus the processor takes an amount of time to produce an efficient object program which is linearly proportional to the number of instructions in the program, NOT to the square of the number as previously. Obviously, when the interaction of instructions over the entire program is considered (as in the Fortran processor) or when the formulae are exceptionally long, this method becomes more valuable as the programs grow larger.

ON PROGRAMMING OF ARITHMETIC OPERATIONS

A. P. ERSHOV

Doklady, AN USSR, vol. 118, No. 3, 1958, pp. 427–430

Translated by MORRIS D. FRIEDMAN, Lincoln Laboratory*

The concepts used without explanation are taken from [1].

1°. Programming algorithms of arithmetic operations (AO) consist of three parts.

The first part A1 successively generates the commands of the AO program.

The second part A2 generates a conventional number (CN)* for each command constructed, which denotes the result of the programmed operation, and replaces it in the formula of the programmed expression. Identification of the entries of similar expressions in the AO formula is made during the A2 operation so that similar expressions will not be programmed repeatedly (economy of command).

The third part A3 replaces the CN in the constructed program, which denotes the intermediate results, by a code of operating registers (OR). New principles for constructing the algorithms of A2 and A3 are proposed herein.

2°. Assumptions and Definitions. Programming of arithmetic operations is carried out on a three address computer. The left side of the AO formula is the superposition of binary and unary operations, each of which is realized by a single command. Each command has one bit σ, which neither enters into the operation code nor into the address part of the command. The A1 algorithm generates the AO commands in the form

a	b	c	σ	Θ

where Θ is the operation code, a and b are CN which denote components and c is a CN which denotes the result (if Θ is a unary operation, then $b = 0$; $c \neq 0$ only for a resultant command, i.e., the concluding command of the calculation of the AO formula; the content of the digit σ is zero at first). A *Block of Preparatory Operations* (BPO) is a group of n registers with the addresses $L+1, \ldots, L+n$ in which are located the AO commands which are generated by the algorithm A1. A *Block of Resultant*

*Translator's note: Apparently, the author means a number given meaning by certain agreed upon conditions. The translator is grateful to Sheldon Best of MIT for having read the translation and making corrections.

[1] A. P. Ershov: Programming programs for the BESM, Moscow. 1958.

Commands (BRC) is a group of registers in which are located all the resultant AO commands in succession. A conventional number of the first kind means a quantity or constant in the formula. A conventional number of the second kind is an intermediate result in the calculation of the formula. It is generated by the algorithm A2 and equals the address of the non-resultant command in the BPO for each such command. A BPO scale is a group of consecutively located registers of the memory with continuous enumeration of the digits, with which the s-th digit of the BPO scale corresponds to the $L+s$ register of the BPO. The scale of the CN of the first kind has a similar apparatus. A *Block of Operating Registers* (BOR) is a group of registers with the addresses $r+1, \ldots, r+m$, where $r+1, \ldots, r+m$ are codes of the operating registers. A *Block of Preparatory Programs* (BPP) is a group of registers in which a preparatory AO program will be placed. The symbol (T) denotes the content of the register T.

3°. In existing command economy methods, the total time of operation of the A2 algorithm is proportional to the square of the number of commands in the AO program.

Shown on figure 1 is a diagram of the A2 algorithm which permits the realization of command economy within a time proportional to the number of commands in the AO program. The basis of the algorithm proposed is the assumption that there exists a certain integer function $F = F(\Theta,a,b)$ $(L+1 \leq F \leq L+n)$ defined for any AO command

a	b	c	σ	Θ

Operation of algorithm A2 is started after construction of the next AO command K by the A1 algorithm (for simplicity, an A2 algorithm is described which does not produce economy of the resultant commands).

Operation 1 investigates whether the command K is the resultant (if not, do operation 2).

Operation 2 calculates $F(\Theta,a,b)$ for the command K and directs the result into the register S. It is evident that $L+1 \leq S \leq L+n$. Let $S = L+p$.

Operation 3 verifies whether $(L+p)$ equals zero (if not, do operation 4).

Operation 4 verifies whether the operation codes, the first two addresses and the digit σ agree for the K and $(L+p)$ commands (if yes, output I).

Operation 5 increases p by one if $p < n$ and puts $L+1$ into the register S if $p = n$.

Operations 6–8 perform if the command K is not economized.

Operation 6 investigates the CN from the address part of the command K. If a CN of the first kind is among them, ones are put in the appropriate digits of the scale of the CN of the first kind and in the p-th digit of the BPO scale (zeroes are in all the digits of both scales before the start of the AO programming).

Operation 7 calculates certain quantities needed for the operation of the A3 algorithm for the command K (see 5°).

Operation 8 directs the command K into the $L+p$ register.

Operation 9 performs if K is a resultant command $(c \neq 0)$. If a one is in the digit of the scale of the CN of the first kind corresponding to the CN c, then BPO commands containing a CN of the first kind are scanned by using the BPO scale. Commands containing the CN c are marked by ones in the σ digit. Consequently, none of the commands containing the CN c in the address part and having been constructed after K will coincide with any of the commands constructed before K during the execution of operation 4. Then K is transmitted to the next free BRC register.

The A2 diagram has two outputs I and II. A CN of the second kind which denotes the result of a constructed nonresultant command is obtained in the register S at the output I. The output II corresponds to a resultant command.

4°. The duration of the execution of A2 is determined by the number of repetitions of operations 3–5. This number depends on the distribution of the values of $F(\Theta,a,b)$ in the strip $[L+1, L+n]$. [Let us note that the usual command economy algorithms correspond to $F(\Theta,a,b) \equiv L+1$.] Evidently, the most favorable case is the uniform distribution of the $F(\Theta,a,b)$ values in $[L+1, L+n]$ for a random composition of the AO formula. In this case, the mathematical expectation Φ of the number of repetitions of the operations 3–5 can be calculated as a function of the number of commands k making up the program and of the quantity of registers n in the BPO [$\Phi = \Phi_n$ (k)]. A derivation of analytic estimates appears to be difficult and the values of Φ_n (k) were computed by the Monte Carlo method. Presented on figure 2 are the curves of $\log_{10} \Phi_n(k)$ obtained for n = 150 (50) 450. Curves of $\log_{10}$ k and $\log_{10} \frac{k^2}{2}$ are given for comparison. It follows from an analysis of the results obtained that, in practice, not more than one execution of the 3–5 operations will occur in each AO command if the BPO exceeds the number of commands in the AO by not less than one and one-half times for all n.

The simplicity of the calculation and the sufficiently uniform distribution of the values is the unique criterion limiting the choice of F(Θ,a,b). It is expedient to use the methods of producing uniformly distributed pseudo-random numbers for the actual construction of F(Θ,a,b). An investigation of the statistical structure of the formulas of the AO to be programmed is of great value in the successful choice of F(Θ,a,b).

5°. There are definite relations in the order of the performance of the operations entering into the AO formula. These relations are given by a rule that the components are calculated, at the beginning, for the components of the formula and then the operation itself is calculated. Consequently, the formula can be considered as a semi-ordered set of the operations therein. Ordering of the operations, caused by the successive location of the command in the program, occurs in the construction of the program to calculate the formula, consequently, the problem of programming the formula can be formulated as a problem in ordering the operations of the formula by retaining a given semi-order. It is evident that the quantity of OR required to calculate the formula depends on the method of ordering its operations. For example, in order to calculate the formula

$$ab + (cd - ef(gh + ij(kl - mn))) \rightarrow y$$

seven OR are needed to perform the action from left to right while only two OR are needed if the calculation is started with the innermost parenthesis. In this connection, the problem arises of finding such an admissible ordering of the operations of the formula for which the minimum quantity of OR would be required for its calculation.

The problem posed is solved partially by using the algorithm A3 of the ordering of the operations of the formula whose diagram is presented on figure 3. Calculation of the operation 7 of the A2 algorithm for each nonresultant command K of two integer functions whose values are put in the third address of the command before it is transmitted to the BPO is preparatory to the operation of A3. The first function, a function of the order P(K), is given by an inductive definition:

A) If the command K does not contain a CN of the second kind, then P(K) = 1.

B_1) If a CN of the second kind, denoting the result of the command K_1 is an address of the command K, then $P(K) = P(K_1)$.

B_2) If CN of the second kind, denoting the result of the commands K_1 and K_2, are in the first and second addresses of the command K, then

$$P(K) = \begin{cases} \max\{P(K_1)P(K_2)\} & \text{if } P(K_1) \neq P(K_2) \\ P(K_1) + 1 & \text{if } P(K_1) = P(K_2) \end{cases}$$

The second function, the entry counter, is calculated as follows. When a command K is transmitted to the BPO, its entry counter equals 0. If a command K′, containing a CN which denotes the result of the command K, is then transmitted to the BPO, then 1 will be added to the entry counter of the command K.

The algorithm A3 starts to perform after the termination of the operation of A1 and A2.

The operation 1 transmits the next AO resultant command into the register R, starting with the last register of the BRC. Let the command K be in R.

Operation 2 replaces the CN of the second kind in K by a code of OR. If a CN of the second kind L+s enters into K, the content of the L+s register is investigated. The command K′ from L+s is transmitted to the first free register of the BOR. The command K′ in L+s is replaced by the address r+i, which indicates where K′ was transmitted to. If K′ has already been replaced by the r+i address during the processing of one of the preceding AO commands, 1 is subtracted from the entry counter of the command K′ in r+i. The CN L+s in K is replaced by the r+i OR code. If two CN of the second kind $L+s_1$ and $L+s_2$ enter into K, where the commands K_1 and K_2 are in the $L+s_1$ and $L+s_2$ registers, that one of the commands K_1, K_2 is transmitted first into the BOR for which the value of the order function is larger.

Operation 3 transmits K to the next register of the BPP, starting with the last register.

Operation 4, scanning from the end of the BOR, finds the first command with entry counter equal to 1. If such a command is not found in the BOR or if no commands are in the BOR, control is transferred to operation 6.

Operation 5 transmits the command found from the r+j register into the R register, puts r+j into the third address of this command and then clears the r+j register.

Operation 6 transfers control to operation 1 if not all the commands are transmitted from the BRC.

The algorithm described solves completely the problem of the most favorable ordering for an AO for which the entry count of each command is 1. This follows from the following two statements which are valid under the above-mentioned limitations:

1. In the interests of the minimum expenditure of operating registers for any binary operation, it is first necessary to calculate those of its components for which the minimum number of OR required for its calculation is larger.

2. The order function for each command equals the minimum quantity of OR required to calculate the expression in which the last operation is realized by the given command.

Moscow University June 27, 1957

AUTOMATIC PROGRAMMING SYSTEMS

Dear Mr. Bemer,

The attached note enumerates some corrections to my short paper in the May issue of the ACM Communications. The error necessitating these corrections was pointed out to me by John M. Brill in a personal communication, and a copy of my reply to him is also enclosed for your information.

I would appreciate it if you would arrange for these corrections to appear in a forthcoming issue of the Communications.

Sincerely yours,
William H. Kautz
Stanford Research Institute

Correction to "Binary and Truth-Functional Operations on a Decimal Computer with an Extract Command" by William H. Kautz (ACM Communications *1* (*5*); 12–13 May 1958)

An error in the description given of the extract command in the above paper has been pointed out. This description should read:

$$aEb=\begin{cases} a \text{ if } a \text{ is even} \\ a+b-1 \text{ if } a \text{ is odd}\end{cases} \text{where } a \text{ and } b \text{ are single decimal digits}$$

In consequence, the inequivalence operation A4 (and the other operations, A6, A7, and A10, which repeat the A4 equation) should be modified as follows:

A4. $X \oplus Y \doteq xE(1^*-y)+yE(1^*-x)$
or $X \oplus Y \doteq (x+y)-2(xEy)$
or $X \oplus Y \doteq (x+y)-(x+y)E0^*$

Also, for no more than 9 variables, $X \oplus Y \oplus Z \oplus \ldots \doteq (x+y+z+\ldots)-(x+y+z+\ldots)E0^*$

Similarly, in Section B, Item 2, the equation and example may be replaced by:

$$X \doteq Y \doteq [(x \doteq y) - (x \doteq y)E0^*]E9^*$$

```
             101001 ───────→  909009
        Add  001011 ───────→  009099
                              ──────
                              918108‡
                  Extract on  000000
                              ──────
                              808008

                              ──────
            Subtract from ‡   110100
                  Extract on  999999
                              ──────
             110100 ←───────  990900
```

I am grateful to Mr. Brill for calling this correction to my attention.

Dear Bob:

Perhaps some of your readers would be interested in knowing that ALGAE I is ready for distribution to interested 704 users. It exists for two 704 configurations, the first of which is the fastest in operation They are:

(a) 32K memory, no drums and 6 tapes (including a combined FORTRAN-ALGAE tape, a card-to-tape formed input tape, and an output tape for off-line printing).

(b) 8K memory, 4 drums and 7 tapes.

It is possible in both systems to perform the compiling with one fewer tape by using the on-line card reader. Both systems write their output on an output tape, hence, a tape-to-printer (or at least a simulator) is required.

Requests for decks or further information should be addressed to:

Mr. Karl Balke
University of California
Los Alamos Scientific Laboratory
Los Alamos, New Mexico

Sincerely yours,
Edward A. Voorhees
Los Alamos Scientific Laboratory

Dear Mr. Bemer:

I am submitting a short note which points out the unappreciated fact that the secant modification of Newton's method is frequently faster on a computer than the classical technique. This note was prompted by J. H. Wegstein's note on "Accelerating Convergence of Iterative Processes," which appeared in the June issue. Although the author does not specifically say so, his method is essentially the secant modification. I have added a footnote which corrects an error in the note of Wegstein.

To clarify a matter of convergence which apparently was overlooked, a technique was discussed in Wegstein's note which is superior to the simple iteration

$$(1) \qquad x_{n+1} = f(x_n)$$

Denoting the error by $\epsilon_n = x_n - x$, where $\dot{x} = f(x)$, one can show that for small errors $\epsilon_{n+1} \cong \epsilon_n \cdot f'(x)$. Hence the four cases are obtained as follows:

$$(2) \qquad \begin{array}{rl} f'(x) < -1 & \text{Oscillatory Divergence} \\ -1 < f'(x) < 0 & \text{Oscillatory Convergence} \\ 0 < f'(x) < 1 & \text{Monotonic Convergence} \\ 1 < f'(x) \phantom{< -1} & \text{Monotonic Divergence} \end{array}$$

Wegstein fails to remark that a converges to $f'(x)$ and hence q converges to $f'(x) \div (f'(x) - 1)$. From this and (2) the table given in section 6 may be obtained.

T. A. Jeeves

COMPUTER	IN LIBRARY	DO NOT HAVE
704	AFAC CAGE CORBIE FORTRAN NYAP PACT 1A REG-SYMBOLIC SAP	ADES FORC KOMPILER 3
701	DUAL-607 FLOP JCS-13 KOMPILER 2 QUICK SHACO SPEEDCODING 3	BACAIC DOUGLAS GEPURS LT-2 PACT I QUEASY SEESAW SO 2 SPEEDEX
705	ACOM AUTOCODER ELI PRINT I SOHIO SYMB. ASSEMBLY	FAIR
702	AUTOCODER ASSEMBLY SCRIPT	
650	ADES II BELL BELL L2, L3 CASE SOAP III DRUCO I EASE II ELI FAST FOR TRANSIT IT RELATIVE SIR SOAP I SOAP II	BACAIC BALITAC ESCAPE FLAIR MITILAC OMNICODE SPEEDCODING SPUR
NORC	NORC COMPILER	
1103- 1103A	CHIP FAP FLIP-SPUR MISHAP RAWOOP-SNAP UNICODE USE	COMPILER I TRANS-USE
UNIVAC I, II	A0, A1, A2 ARITHMATIC (A3) BIOR FLOWMATIC (B-0) GP GPX (II ONLY) MATHMATIC (AT3) MATRIX MATH NYU, OMNIFAX SHORTCODE X-1	MJS RELCODE
D'TRON 201 204 205	APS III PURDUE COMPILER UGLIAC	DATACODE I DUMBO SAC STAR
UDEC III		UDECIN-1 UDECOM-3
G-15	DAISY 201 INTERCOM 101 INTERCOM 1000	
WHIRL- WIND	COMPREHENSIVE SUMMER SESSION	ALGEBRAIC
FERUT	TRANSCODE	
JOHNNIAC	EASY FOX	
ILLIAC	ILLIAC	
LGP-30	ERFPI JAZ SPEED	
MIDAC	EASIAC MAGIC	

SECANT MODIFICATION OF NEWTON'S METHOD

T. A. JEEVES, Westinghouse Research Laboratories

The use of a secant approximation to the derivative in Newton's iterative scheme for finding the roots of equations produces a procedure which, generally, takes less computer time than the classical technique. Although the secant method requires more iterations, each iteration requires less time since there is no evaluation of the derivative of the function. The secant method also has the advantage of needing less memory space.

Newton's iterative scheme for finding a root of $f(x)=0$ is

$$(1) \qquad x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}$$

Denoting the error by $\epsilon_n=x_n-x$, where x is the root, then the error is, for small errors,

$$(2) \qquad \epsilon_{n+1}\cong k\,\epsilon^2_n$$

where $k=1/2\,f''(x)\div f'(x)$. Now the number of significant figures in the answer is to an additive constant, $d_n=-\log \epsilon_n$. Applying this to (2)

$$d_{n+1}\cong 2d_n-\log k,$$

or expressing this by differences

$$(3) \qquad \Delta d_{n+1}\cong 2\,\Delta d_n,$$

where $\Delta d_n=d_n-d_{n-1}$. The relation (3) says that at each iteration the increase in the number of significant figures is double the previous increase. (This relation (3) is commonly, if inaccurately, described by saying the number of figures is doubled at each step.) The above scheme will be called the tangent method.

The secant modification consists of replacing the derivatives by a difference quotient, so that

$$\frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}} \text{ replaces } f'(x_n).$$

Then the iteration takes the form

$$(4) \qquad x_{n+1}=\frac{x_{n-1}\,f(x_n)-x_n\,f(x_{n-1})}{f(x_n)-f(x_{n-1})},$$

and the error is, for small errors,

$$(5) \qquad \epsilon_{n+1}\cong k\,\epsilon_n\,\epsilon_{n-1},$$

where again $k=1/2\,f''(x)\div f'(x)$. To obtain the number of significant figures in the answer the difference equation (5) is transformed into an equation like (2) involving only two index values, $n+1$ and n. This is done by first solving (5) to yield

$$\epsilon_n\cong\frac{1}{k}R^{\rho^n},$$

where $\rho=1/2\,(1+\sqrt{5})\cong 1.62$, and R is a constant. Second, a new difference equation is formed in which R does not appear:

$$\epsilon_{n+1}\cong\frac{1}{k}R^{\rho^{n+1}}=\frac{1}{k}\left(R^{\rho^n}\right)^{\rho}\cong\frac{1}{k}\left(k\,\epsilon_n\right)^{\rho}=k^{1-\rho}\,\epsilon^{\rho}_n,$$

that is,

$$\epsilon_{n+1}\cong k^{1-\rho}\,\epsilon^{\rho}_n.$$

Now proceeding as in the preceding paragraph (see (3)) one obtains:

$$(6) \qquad \Delta d_{n+1}\cong\rho\cdot\Delta d_n.$$

Thus at each iteration the increase in the number of significant figures is 1.62 times the previous increase.[1]

The relative computer speeds of the tangent method and the secant method will now be compared. The number of iterations required in the two variations is, from (3) and (6), in the ratio of one to $\log 2 \div \log \rho = 1.43$. In the tangent method the derivative as well as the function must be evaluated at each point. If the derivative calculation requires an amount of time which is w times the time to evaluate the function, then the total time in the two methods is in the ratio $1 + w$ to 1.43. (Here the time to carry out (1) or (4) is negligible in comparison to the time required to evaluate the function.) Hence, if $w > .43$, that is, if the time to evaluate the derivative exceeds 43 per cent of the time to evaluate the function, then the secant method will require less computer time. In practice, w is generally about 1, and the secant method is superior.

The secant method is certainly not new; however, the observation that it can actually produce a faster computer program than the classical tangent method seems to have escaped attention.

[1] Not a factor of 2 as Wegstein asserts, hence the convergence is *not* quadratic.

RECURSIVE CURVE FITTING TECHNIQUE

John Giblin, Standard Oil Co. of Ohio, Cleveland, Ohio

INTRODUCTION

The idea of recursive curve fitting has been in use for some time[1] as a graphical technique for fitting curves "by eye" to observational data thought to be a function of two or more variables. The purpose of this note is to point up the potentialities of applying the recursive scheme. It will be seen that such a scheme is applicable when least square techniques are available for determining subsets of the unknown curve or surface parameters.

METHOD

Consider the following problem:

Given the form

(1) $$Y = AX + B + (CZ + D)^{-1} + E,$$

where E is the error term, determine A, B, C, and D so as to minimize

$$\Sigma\,[Y - (AX + B) - (CZ + D)^{-1}]^2$$

Suppose for the moment that techniques are available for (a) determining A and B so as to minimize ΣE_1^2, where

(2) $$E_1 = Y - (AX + B)$$

and (b) for determining C and D so as to minimize ΣE_2^2, where

(3) $$E_2 = [E_1 - (CZ + D)^{-1}]$$

Let A_1, B_1, C_1, and D_1 be the values so obtained. Then it can be shown that the process of finding corrected values of the above constants by means of operating on the E's shown below converges at a relative minimum in the residual sum of squares. The operators are the techniques a and b, described above. The E's are as follows:

(4) $E_3 = E_2 - (A_1X + B_1)$ (Operate on E_3 with technique a)

(5) $E_4 = E_3 - (C_1Z + D_1)^{-1}$ (Operate on E_4 with technique b)

and in general

[1] Methods of Correlation Analyses by Ezekiel, John Wiley and Sons, Chapters 16 and 22.

(6) $E_i = E_{i-1} - F_i,$

where $i = 3, 4, \ldots\ldots$ and

(7) $F_i = \frac{A_{i-1}}{2}X + \frac{B_{i-1}}{2}$ (When i is odd)

(8) $F_i = \left[\frac{C_i Z}{2} + \frac{D_i}{2}\right]^{-1}$ (when i is even)

Techniques a and b are to operate on even and odd E's respectively.

In the problem under consideration the usual least squares technique applied directly to eq. (1) leads to insoluble equations. In general, breaking the problem into two parts, as shown above, often leads to easy application of the least squares process to each part separately. Then the repeated application of techniques corresponding to a and b to residuals given by eq. (6), leads to a convergent solution.

Example:

The solution corresponding to eq. (1) has been programmed for the IBM 650. The method used to determine C and D was developed by writing

(9) $(CZ+D)^{-1} = C^{-1}\ (Z+D/C)^{-1} = W/C,$

where

(10) $W \equiv (Z+\alpha)^{-1};\ \alpha = D/C$

Now let α be fixed and determine $\frac{1}{C}$ so as to minimize ΣE_i^2, where

(11) $E_i = Y_i - W/C$

Differentiation gives

(12) $\frac{1}{C} = \frac{\Sigma WY}{\Sigma W^2}$

and substitution of this value into eq. (11) results in

(13) $\Sigma E_i^2 = \Sigma Y^2 - \frac{(\Sigma WY)^2}{\Sigma W^2}$

Equation 13 shows that no matter what α is chosen, a non-negative reduction in ΣY^2 is effected. However, there are optimum values of α, and the 650 computer program uses a binary search about an automatically selected α_0 to determine such an optimum. The value of α_0 is chosen by selecting both C and D so that $(CZ+D)^{-1}$ passes through two of the Y values.

CONCLUSIONS

The automatic selection of the optimum form of equation to be used in representing a set of observational data would prove a valuable acquisition at almost every computer installation. The general technique presented here can be used to fit a wide variety of forms, from which the most applicable can be chosen.

Editor's Note: The following article will be published in two parts. Part 2 will appear in a later issue of the COMMUNICATIONS.

THE PROBLEM OF PROGRAMMING COMMUNICATION WITH CHANGING MACHINES

A PROPOSED SOLUTION

Report of the Share Ad-Hoc Committee on Universal Languages

By J. Strong, North American Aviation
J. Wegstein, Bureau of Standards
A. Tritter, Lincoln Laboratory
J. Olsztyn, General Motors Corp.
O. Mock, North American Aviation
T. Steel, Systems Development Corp.

I. *BASIC ASSUMPTIONS.*

One of the fundamental problems facing the computer profession today is the considerable length of time required to develop an effective method of communication with the machine. Moreover, it seems that the ability to communicate easily is no sooner acquired than the language changes, and the problem is renewed, usually at a higher level of complexity.

A. *Obsolescence of Machines.* Experience would indicate that there are few machine users who do not obtain new and different machines every three to five years—a change generally prompted by technical obsolescence rather than by the decaying ability of the machine. In most cases, the appearance of a new machine coincides with the necessity to expand computing capacity. Although the programming cost involved in changing machines is high, the pressure to advance to these new machines has been sufficiently great to justify their acquisition.

B. *Growing Sophistication of Machine Languages.* So far, each advance in machine design has been accompanied by an increased complexity in the structure of its language, making programming in machine-like language progressively more costly in both dollars and elapsed time.

C. *Existing Compilers.* Current compilers alleviate the problem somewhat by translating some particular "easy-to-code" language into a specific machine code. The principal shortcoming in this approach is the considerable lapse of time between the initial conception of the "easy-to-code" language and its general acceptance. This time is of the same order of magnitude as the machine replacement cycle, resulting in the ever-present danger of having a good compiler available for the old machine just after it has been replaced.

II. *APPROACH TO A SOLUTION.*

Let us examine a broad set of specifications which might describe a "system" capable of solving this problem.

A. *Ideal Specifications.*

1. The individual with a problem would describe its method of solution in the language most natural to his way of thinking about the problem.
2. Once his problem has been coded in this language, the solution might be obtained by processing the program with acceptable efficiency on any present or future computer. (Efficiency of operation in a proposed system is equal to the lowest possible total cost, including programming, divided by actual total cost.)
3. A minimum of "system programming" should be required to produce the system initially and to maintain it. Most installations should not be required to do any system programming at all.

B. *Practical Specifications.* The ideal specifications listed above need some interpretation. In their ultimate implications, they may be incapable of realization in this century. A few qualifications are in order.

1. Saying that the man with the problem should speak in the "most natural" language is misleading. His native tongue, English, is capable of almost infinite shades of meaning. Few people can use English precisely and unambiguously. Consequently, the most natural method of expression will have to be compromised in the interest of precision. This is not unusual; almost every scientific discipline, and most trades and professions, have their own unique, fairly precise language.
2. The ideal of processing every problem in every language on every computer and producing efficient results can probably be realistically approached without too much compromise by permitting the following conditions:
   a. The coder can achieve higher efficiency if he knows which specific computer his routine will be processed on.
   b. Any routine written for a small computer may be processed on a larger one. However, a routine written for a larger computer might be capable of execution on only a few small computers and certainly not on a computer several orders of magnitude smaller. For example, 709 routines would not normally be translated into E101 language since they might run for many years on the smaller machine.
   c. A large computer is available for development of the system. A run on a large computer would prepare the system for all subsequent use on a small computer.
   d. Although the existence of the system may influence computer design, the system should not be dependent on this factor. Moreover, it would be undesirable if the system should in any way inhibit the development of more powerful computers by restricting the complexity of their basic languages.
3. With regard to the time required for system programming, one would hope that some useful results of the system would be available in from three to five years. The design of the system should not rely on a major "breakthrough" in programming art. The system, for example, should not stand or fall upon its ability to develop a routine which is so general that it can transform every type of language into every other type.

III. *FURTHER ASSUMPTIONS.*

The proposed solution to the problem will be presented in some detail in paragraph 5 below. Since some of the assumptions underlying this solution may be considered controversial, a preliminary discussion of them here may help clarify the final proposal.

A. *Languages.*

1. It is impossible to agree on one universal POL. Since there are many varieties of problems, any attempt at universality of problem-oriented languages will result either in inadequacy (such as an attempt to use algebraic language for a logical problem) or such extensiveness as to become useless. In the latter case, the "universal" POL is really the sum of all possible POL's and is never truly universal for long, since the language must grow to cope with the new classes of problems that arise.
2. Machine languages will continue to grow in complexity and will become increasingly difficult to code in. Everyone looks with dread at the possible computers of the next decade, which will be simultaneously executing multiple asynchronous stored programs. There is little reason to expect a reversal of this trend.
3. The present status of the compiler art requires a rather difficult generative routine to transform each POL formulated into each ML desired. Moreover, additional routines must be written whenever it is necessary to produce a different ML from the one belonging to the machine on which the translation routine is executed. The number of individual compilers of the current type needed can only increase as it becomes desirable for POL's to multiply, for machines to be replaced, and for one organization to have several types of machines. The time is fast approaching when the need for these routines will exceed any possible supply.

B. *Programming.*

1. At the present state of the programming art, a reasonably efficient routine can be written to transform any one specific language into any other specific language.
2. The complexities of the language transformed determine the size of the machine needed for an acceptable degree of efficiency in both the transformation process and the subsequent execution of the ML routine.

C. *Machines.* There will be available for the system programmers a machine at least as complex as the IBM 709 and machine language programming tools at least as versatile as the SHARE 709 programming system.

IV. *SOLUTION—THE THREE-LEVEL CONCEPT ("UNCOL").*

A. *History.* This concept is not particularly new or original. It has been discussed by many independent persons as long ago as 1954. It might not be difficult to prove that "this was well-known to Babbage," so no effort has been made to give credit to the originator, if indeed there was a unique originator.

B. *Outline.* The system is composed of three levels of language, as shown in the schematic diagram, Appendix B.

1. ML Level. The lowest level (closest to the bits in the machine hardware) is composed of all the current or future ML's.
2. POL Level. The highest level (furthest from the machine) is composed of all the current and future POL's.
3. UNCOL Level. The center level is a single language "UNCOL," the Universal Computer Oriented Language.
4. Generators. Generators are those routines which perform the transformation from the POL's to UNCOL. They are analogous to present generative compilers except that they produce, not a number of ML's, but only UNCOL. As with present compilers, one of these would be needed for each POL used with a given machine.
5. Translators. These are routines which perform the transformation from UNCOL to ML, and are like present compilers in that they are one-time preprocessors before execution of the customer's program. However, they are not so complex nor so difficult to write as the "generators" described above, since UNCOL, being computer oriented, has many things in common with each of the ML's. For each machine only one translator need ever be produced, regardless of the number of POL's formulated or used.

V. *CONCLUSIONS.*

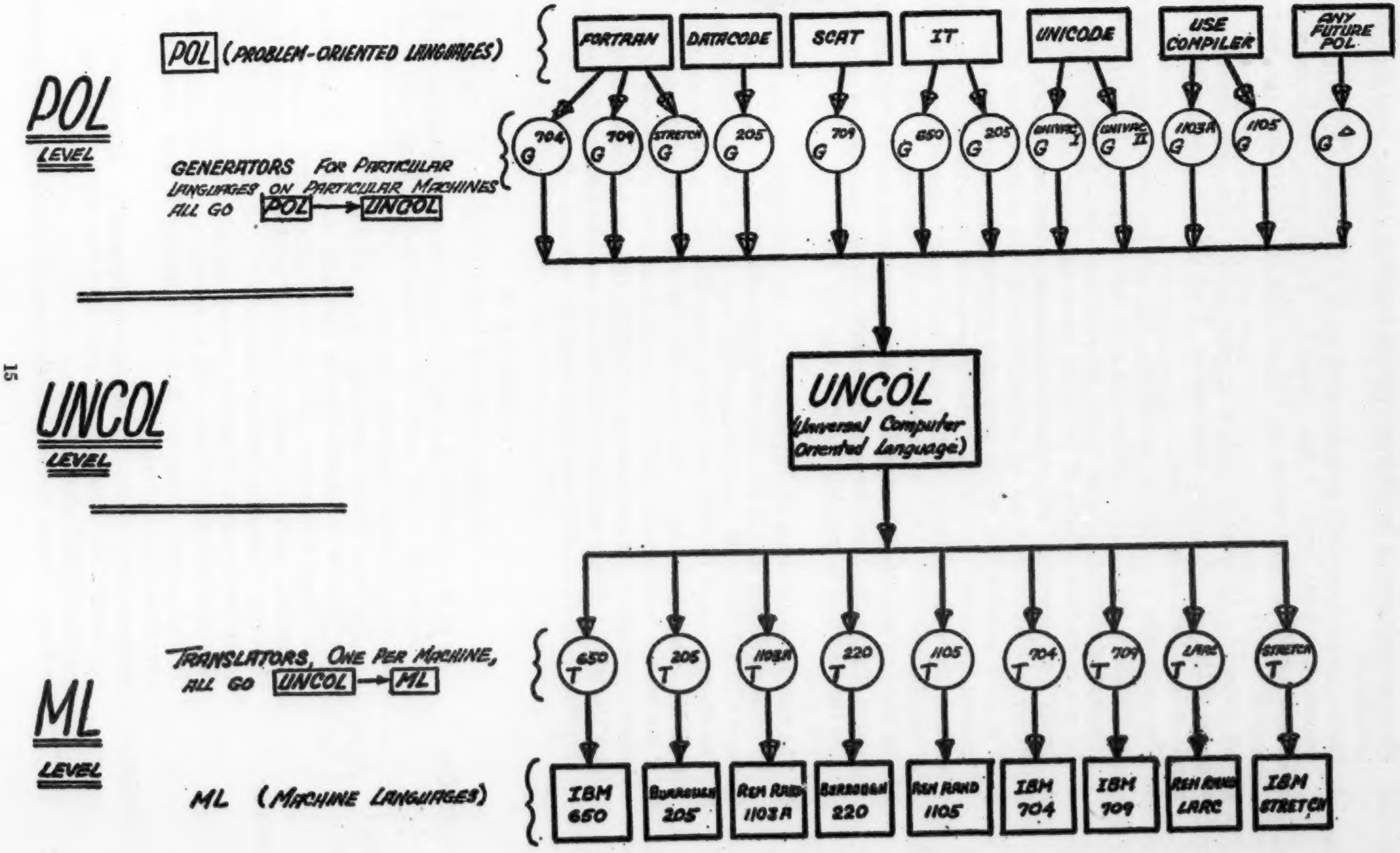
It might seem that the UNCOL three-level system does nothing but complicate the current method, whereby compilers enable one to proceed with direct simplicity from a chosen POL to a specific ML. Such is not the case. The versatility of this system is almost unlimited. Because of its complexity, detailed discussion of UNCOL has been found to require a special flow chart notation (described in Appendix B). With the addition of this notation, illustrative examples of how this system might function are presented in Appendix D. Conclusions are summarized below.

A. *Definite Immediate Benefits.*

1. As each new machine is produced, all that would be necessary in the way of system programming is a single translator to convert UNCOL into the new ML. The machine manufacturer would probably write several successive versions for each of his machines in an attempt to exploit to the fullest the outstanding features of each machine.
2. As each POL is invented, a system programmer must write a generator to translate his POL into UNCOL. *Thereafter, his POL will never become obsolete.* Routines written in it can be executed on any machine at any time in the future.
3. The UNCOL system will enable an installation to use its current programs on any machine without additional programming cost.
4. The UNCOL system can be designed, programmed, and installed within three years. The

THE 3-LEVEL CONCEPT

2-28-58



above advantages can be achieved without requiring a major breakthrough in programming techniques.

5. System programmers will probably work most of the time in UNCOL with only occasional descents to the ML level. A large percent of the customer's problems will be coded in one or another POL. If, however, no POL is suitable, the problem can be coded in UNCOL directly.

B. *Growth Potential.* Although the following potentialities exist, they are independent of the advantages of the UNCOL system listed above. That UNCOL provides for growth in this direction is an important extra dividend.

1. "Boot-strapping" is a distinct possibility. By boot-strapping is meant the ability to adapt the system to new POL's, machines (or even new versions of UNCOL) with a minimum of human programming. That is to say, the system can be to a large extent self-renewing, with the system routines capable of producing newer and better system routines.
2. Simple "boot-strapping" will probably be available almost immediately. System programmers writing in UNCOL can use an existing translator to produce their ML system programs.
3. It is quite possible that within five years programming techniques may permit writing a "general translator" in UNCOL which, given the characteristics of machine "A", would produce (on other machines) a translator that would convert from UNCOL into the ML of machine "A".
4. The last and least probable step would be the development of the "general generator". This routine, if given the characteristics of the POL and the machine it was to run on, would produce the generator to translate the POL into UNCOL.

C. *UNCOL Itself.*

1. The first step must be the development of UNCOL and its acceptance as a universal standard by some significant part of the computing profession.
2. Since UNCOL would be computer-oriented, any POL could eventually be expressed in it.
3. The effectiveness of UNCOL will depend upon how easy it will be to translate from it into each ML, while exploiting at the same time the advantages of the new machine concerned.
4. If the scheme is successful, UNCOL I should have a life expectancy of ten to fifteen years before any large revision is necessary. UNCOL II could be devised with the "general generator" in mind. Any transition to UNCOL II could be accomplished everywhere by bootstrapping techniques. Only one routine need be written.

APPENDIX B
UNCOL SYSTEM NOTATION

LANGUAGES refer to the symbols and rules for using them *as understood by a human being.* This is independent of the particular vehicle (and its local code) used to carry them (e.g. punched cards, magnetic tape, etc.).

POL: Any *P*roblem-*O*riented *L*anguage, e.g. FORTRAN.

ML: Basic *M*achine *L*anguage for any particular machine. If accompanied by appropriate "input translators", ML can be extended to include "machine-like" languages unique to a particular machine, e.g. SAP symbolic language for the 704.

UNCOL: *UN*iversal *C*omputer-*O*riented *L*anguage. This is a standard language oriented to the requirements of general purpose digital computers. It must have at least their characteristics of being able to express any computable problem.

(In the general schematic diagram, Appendix B, and there only, languages are surrounded by a rectangle ▭).

ROUTINES are sets of instructions arranged in proper sequence. They can exist in any language, and usually are transformed at least once into another language. They are designated by capital letters, thus:

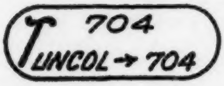
P: *Problems.* Those routines which express the solution to a problem. Normally these are "customer's jobs" (e.g. payroll, aircraft performance, etc.) and can most easily be written in a POL.

G: *Generators.* Those routines which operate on routines existing in a POL and perform the transformation from one specific POL to UNCOL.

T: *Translators.* Those routines which operate on a routine existing in UNCOL and perform the transformation from UNCOL to one specific ML.

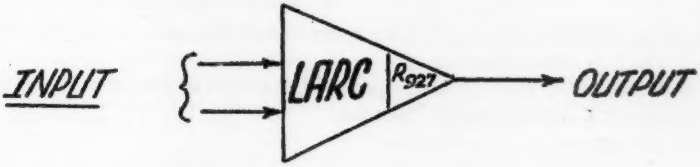
SUPERSCRIPTS The symbol for a routine carries a superscript, indicating the specific language in which the routine exists, e.g. FORTRAN (meaning that particular POL), UNCOL (which is unique), or 704 (meaning that particular ML).

SUBSCRIPTS are used to designate the operation the routine performs. Normally there is no need to use a subscript with P. However, G and T *must* have a subscript, denoting the transformation that the routine performs. For example; the notation:



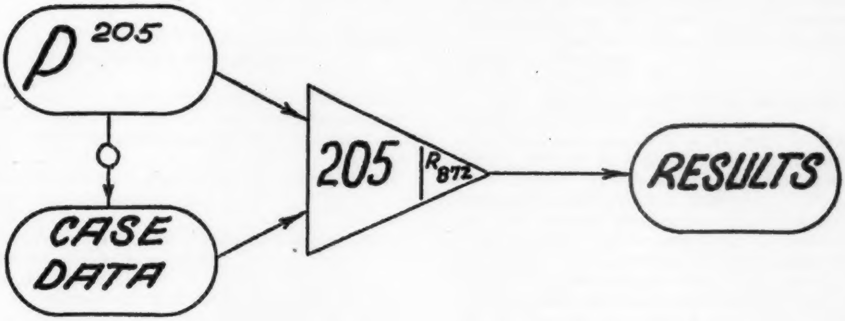
denotes a translator which transforms a routine in UNCOL into the same routine in 704 machine language. The translation routine itself is in 704 language.

MACHINES are denoted by a triangle, thus:



The number, preceded by R, at the apex denotes a specific machine run.

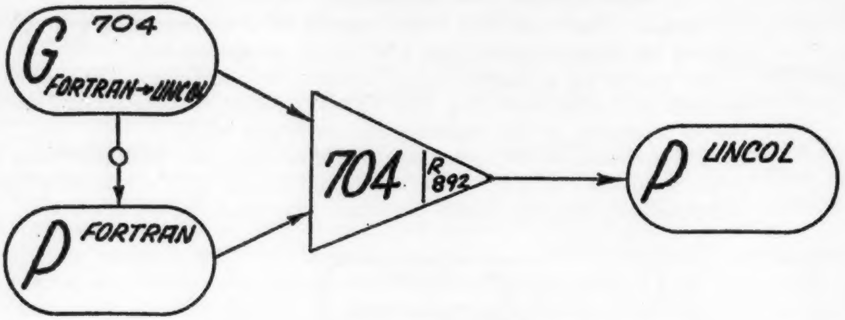
FLOW CHARTS indicate a machine run. All necessary routines and data are shown as input. The output is also defined. Routines are shown inside the oval symbol: ○. For example, a conventional production run, #872, on an Electrodata 205 is shown as:



The symbol ○ means "operating on". Therefore, the above chart means:

"A Problem routine in 205 language and its CASE DATA go into the 205 and appropriate RESULTS are produced.

Generators and translators operate on other routines (usually Problem routines) and produce the routine in another language as output; for example, a 704 run #892 might be:



This means: "The Generator in 704 language operates (in the 704) on the Problem in FORTRAN language and produces a Problem routine in UNCOL language as output."

NOTE several rules which are useful:

(a) The routine which is being executed must have a *superscript* the same as the machine being used for the run.

(b) The routine being operated upon must have a *superscript* the same as the *first half of the subscript* of the G or T which is being executed.

(c) Output is the routine that was operated upon with its *superscript* the same as the *last half of the subscript* of the G or T which is being executed.

STANDARDS

The following is a continuation of the Standard Department's publication of the "Glossary of Computer Engineering and Programming Terminology" from the Aberdeen Proving Ground, *BRL Report No. 1010*.

The department earnestly solicits all comments concerning the proper use of terms, definitions, ambiguities and unusual meanings and applications.

CHECK, MATHEMATICAL or ARITHMETICAL

a check making use of mathematical identities or other properties, frequently with some degree of discrepancy being acceptable; e.g., checking multiplication by verifying that $A \cdot B = B \cdot A$, checking a tabulated function by differencing, etc.

CHECK, MODULO *N*

a form of check digits, such that the number of ones in each number A operated upon is compared with a check number B, carried along with A and equal to the remainder of A when divided by N, e.g., in a "modulo 4 check", the check number will be 0, 1, 2, or 3 and the remainder of A when divided by 4 must equal the reported check number B, or else an error or malfunction has occurred; a method of verification by congruences, e.g. casting out nines.

CHECK, ODD–EVEN

a check system in which a one or zero is carried along in a word depending on whether the total number of ones (or zeros) in a word is odd or even.

CHECK, PARITY

a summation check in which the binary digits, in a character or word, are added (modulo 2) and the sum checked against a single, previously computed parity digit; i.e., a check which tests whether the number of ones is odd or even.

CHECK, PROGRAMMED

a system of determining the correct program and machine functioning either by running a sample problem with similar programming and known answer, including mathematical or logical checks such as comparing A times B with B times A and usually where reliance is placed on a high probability of correctness rather than built-in error-detection circuits or by building a checking system into the actual program being run and utilized for checking during the actual running of the problem.

CHECK, REDUNDANT

a check which uses extra digits, short of complete duplication, to help detect malfunctions and mistakes.

CHECK, SUMMATION

a redundant check in which groups of digits are summed, usually without regard for overflow, and that sum checked against a previously computed sum to verify accuracy.

CHECK, TRANSFER

verification of transmitted information by temporary storing, re-transmitting and comparing.

CHECK, TWIN

a continuous duplication check achieved by duplication of hardware and automatic comparison.

CHECKING, MARGINAL

a system or method of determining computer circuit weaknesses and incipient malfunctions by varying the power applied to various circuits, usually by a lowering of the D.C. supply or filament voltages.

CLAMPING–CIRCUIT

a circuit which maintains either amplitude extreme of a waveform at a given voltage level, or potential.

CLEAR

to replace all information in a storage device by ones or zeros as expressed in the number system employed.

CLOCK, MASTER

the source of standard signals required for sequencing computer operation, usually consisting of a timing pulse generator, a cycling unit and sets of special pulses that occur at given intervals of time. Usually in synchronous machines the basic frequency utilized is the clocking pulse.

CLOSED–SHOP

this is intended to mean that mode of computing machine support wherein the applied programs and utility routines are written by members of a specialized group whose only professional concern is the use of computers.

CODE

a system of symbols and their use in representing rules for handling the flow or processing of information; to actually prepare problems for solution on a specific computer.

CODE, COMPUTER

the code representing the operations built into the hardware of the computer.

CODE, EXCESS–THREE

A coded decimal notation for decimal digits which represents each decimal digit as the corresponding binary number plus three, e.g. the decimal digits 0, 1, 7, 9 are represented as 0011, 0100, 1010, 1100, respectively. In this notation, the nines complement of the decimal digit is equal to the ones complement of the corresponding four binary digits.

CODE, INSTRUCTION

an artificial language for describing or expressing the instructions which can be carried out by a digital computer. In automatically sequenced computers, the instruction code is used when describing or expressing sequences of instructions, and each instruction word usually contains a part specifying the operation to be performed and one or more addresses which identify a particular location in storage. Sometimes an address part of an instruction is not intended to specify a location in storage but is used for some other purpose.

If more than one address is used, the code is called a multiple-address code.

CODE, INTERPRETER

a code which is acceptable to an interpretive routine.

CODE, MULTIPLE–ADDRESS

an instruction or code in which more than one address or storage location is utilized. In a typical instruction of a Four-Address Code the addresses specify the location of two operands, the destination of the result, and the location of the next instruction in the sequence. In a typical Three-Address Code, the fourth address specifying the location of the next instruction is dispensed with, the instructions are taken from storage in a preassigned order. In a typical Two-Address Code, the addresses may specify the locations of the operands. The results may be placed at one of the addresses or the destination of the results may be specified by another instruction.

CODE, OPERATIONAL

that part of an instruction which designates the operation to be performed.

CODING

the list, in computer code or in pseudo-code, of the successive computer operations required to solve a given problem.

CODING, ABSOLUTE, RELATIVE or SYMBOLIC

coding in which one uses absolute, relative, or symbolic addresses, respectively; coding in which all addresses refer to an arbitrarily selected position, or in which all addresses are represented symbolically.

CODING, ALPHABETIC

a system of abbreviation used in preparing information for input into a computer such that information is reported in the form of letters, e.g., New York as NY, carriage return as CN, etc.

CODING, AUTOMATIC

any technique in which a computer is used to help bridge the gap between some "easiest" form, intellectually and manually, of describing the steps to be followed in solving a given problem and some "most efficient" final coding of the same problem for a given computer; two basic forms are Routine, compilation and Routine, interpretation.

CODING, NUMERIC

a system of abbreviation used in the preparation of information for machine acceptance by reducing all information to numerical quantities; in contrast to alphabetic coding.

COLLATE

to combine two or more similarly ordered sets of items to produce another ordered set composed of information from the original sets. Both the number of items and the size of the individual items in the resulting set may differ from those of either of the original sets and of their sums, sequence 23, 24, 48 may be collated into 12, 23, 24, 29, 42, 48; to combine two or more sequences of items according to a prescribed rule such that all items appear in the final sequence.

COLLATOR

a machine which has two card feeds, four card pockets and three stations at which a card may be compared or sequenced with regard to other cards so as to select a pocket in which it is to be placed, e.g., the machine is suitable for matching detail cards with master cards, merging cards in proper sequence, etc.

COLUMN

one of the character or digit positions in a positional notation representation of a unit of information, columns are usually numbered from right to left column, zero being the right-most column if there is no point, or the column immediately to the left of the point if there is one; a position or place in a number in which the position designates the power of the base and the digit is the coefficient, e.g., in 3876, the 8 is the coefficient of 10^2, the position of the 8 designating the 2.

COMMAND

a pulse, signal, or set of signals initiating one step in the performance of a computer operation. See instruction and order.

COMPARATOR

a device for comparing two different transcriptions of the same information to verify the accuracy of transcription, storage, arithmetic operation or other process, in which a signal is given dependent upon the relative state of two items, i.e. larger, smaller, equal, difference, etc.

COMPARE

to examine the representation of a quantity for the purpose of discovering its relationship to zero, or of two quantities for the purpose of discovering identity or relative magnitude.

COMPARISON

determining the identity, relative magnitude and relative sign of two quantities and thereby initiating an action.

COMPARISON, LOGICAL

the operation concerned with the determination of similarity or dissimilarity of two items, e.g. if A and B are alike, the result shall be "1" or yes, if A and B are not alike or equal, the result shall be "0" or no, signifying "not alike".

COMPILER

a program making routine, which produces a specific program for a particular problem by determining the intended meaning of an element of information expressed in pseudo-code, selecting or generating the required subroutine, transforming the subroutine into specific coding for the specific problem, assigning specific storage registers, etc. and entering it as an element of the problem program, maintaining a record of the subroutines used and their position in the program and continuing to the next element of information in pseudo-code.

COMPLEMENT

a quantity which is derived from a given quantity, expressed to the base n, by one of the following rules and which is frequently used to represent the negative of the given quantity. (a) *Complement on n:* subtract each digit of the given quantity from *n-1,* add unity to the least significant digit, and perform all resultant carrys. For example, the *twos complement* of binary 11010 is 00110; the *tens complement* of decimal 456 is 544. (b) *Complement on n-1:* subtract each digit of the given quantity from n-1. For example, the *ones complement* of binary 11010 is 00101; the *nines complement* of decimal 456 is 543.

COMPUTER

any device capable of accepting information, applying prescribed processes to the information, and supplying the results of these processes; sometimes, more specifically, a device for performing sequences of arithmetic and logical operations; sometimes, still more specifically, a stored-program digital computer capable of performing sequences of internally-stored instructions, as opposed to *calculators* on which the sequence is impressed manually (desk calculator) or from tape or cards (card programed calculator).

COMPUTER, ANALOG

a calculating machine which solves problems by translating physical conditions like flow, temperature or pressure into electrical quantities and using electrical equivalent circuits for the physical phenomenon.

COMPUTER, ASYNCHRONOUS

a calculating device in which the performance of any operation starts as a result of a signal that the previous operation has been completed; contrasted with synchronous computer.

COMPUTER, AUTOMATIC

a calculating device which handles long sequences of operations without human intervention.

COMPUTER, DIGITAL

a calculating device utilizing numbers to express all the variables and quantities of a problem. The numbers are usually expressed as a space-time distribution of punched holes, electrical pulses, sonic pulses, etc.

COMPUTER, SYNCHRONOUS

a calculating device in which the performance of all operations is controlled with equally spaced signals from a master clock.

CONDITIONAL

subject to the result of a comparison made during computation; subject to human intervention.

CONTENTS

the information stored in any storage medium. Quite prevalently, the symbol ( ) is used to indicate "the contents of"; e.g., (m) indicates the contents of the storage location whose address is m; (A) indicates the contents of register A; (T_2) may indicate the contents of the tape on input-output unit two, etc.

CONTROL

(1) Usually, those parts of a digital computer which effect the carrying out of instructions in proper sequence, the interpretation of each instruction, and the application of the

proper signals to the arithmetic unit and other parts in accordance with this interpretation. (2) Frequently, one or more of the components in any mechanism responsible for interpreting and carrying out manually-initiated directions. Sometimes called manual control. (3) In some business applications of mathematics, a mathematical check.

CONTROL, CASCADE
an automatic control system in which various control units are linked in sequence, each control unit regulating the operation of the next control unit in line.

CONTROL-SEQUENCE
the normal order of selection of instructions for execution. In some computers, one of the addresses in each instruction specifies the control sequence. In most other computers the sequence is consecutive except where a jump occurs.

CONTROL, SEQUENTIAL
a manner of operation of a computer such that instructions are fed in a given order to the computer during the solution of a problem.

CONTROL-UNIT
that portion of the hardware of an automatic digital computer which directs the sequence of operations, interprets the coded instructions, and initiates the proper commands to the computer circuits to execute the instructions.

CONVERT
to change numerical information from one number base to another (e.g., decimal to binary) and/or from some form of fixed point to some form of floating-point representation, or vice versa.

CONVERTER
a unit which changes the language of information from one form to another so as to make it available or acceptable to another machine, e.g., a unit which takes information punched on cards to information recorded on magnetic tape, possibly including editing facilities.

COPY
to reproduce information in a new location replacing whatever was previously stored there and leaving the source of the information unchanged.

CORE, MAGNETIC
a magnetic material capable of assuming and remaining at one of two or more conditions of magnetization, thus capable of providing storage, gating or switching functions, usually of toroidal shape and pulsed or polarized by electric currents carried on wire wound around the material.

COUNTER
a device, register, or storage location for storing integers, permitting these integers to be increased or decreased by unity or by an arbitrary integer, and capable of being reset to zero or to an arbitrary integer.

COUNTER, CONTROL
a device which records the storage location of the instruction word, which is to be operated upon following the instruction word in current use. The control counter may select storage locations in sequence, thus obtaining the next instruction word from the following storage location, unless a transfer or special instruction is encountered.

COUNTER, RING
a loop of interconnected bistable elements such that one and only one is in a specified state at any given time and such that, as input signals are counted, the position of the one specified state moves in an ordered sequence around the loop.

COUPLING
the means by which energy is transferred from one circuit to another; the common impedance necessary for coupling.

COUPLING, CAPACITIVE
a method of transferring energy from one circuit to another by means of a capacitor that is common to both circuits.

COUPLING, DIRECT
a method of transferring energy from one circuit to another by means of resistors common to both circuits.

CRT
cathode ray tube; a device yielding a visual plot of the variation of several parameters by means of a proportionally deflected beam of electrons.

CYBERNETICS
the comparative study of the control and intracommunication of information handling machines and nervous systems of animals and man in order to understand and improve communication, e.g., a study of the art of the pilot or steersman.

CYCLE
a set of operations repeated as a unit; a non-arithmetic shift in which the digits dropped off at one end of a word are returned at the other end in circular fashion; cycle right and cycle left. To repeat a set of operations a prescribed number of times including, when required, supplying necessary address changes by arithmetic processes or by means of a hardware device such as a *B-box* or *cycle-counter*.

CYCLE COUNT
to increase or decrease the cycle index by unity or by an arbitrary integer.

CYCLE-CRITERION
the total number of times the cycle is to be repeated; the register which stores that number.

CYCLE-INDEX
the number of times a cycle has been executed; or the difference, or the negative of the difference, between that number and the number of repetitions desired.

CYCLE, MAJOR
the maximum access time of a recirculating serial storage element; the time for one rotation, e.g., of a magnetic drum or of pulses in an acoustic delay line; a whole number of minor cycles.

CYCLE, MINOR
the word time of a serial computer, including the spacing between words.

CYCLE, RESET
to return a cycle index to its initial value.

DAMPING
a characteristic built into electrical circuits and mechanical systems to prevent rapid or excessive corrections which may lead to instability or oscillatory conditions, e.g., connecting a resistor on the terminals of a pulse transformer to remove natural oscillations; placing a moving element in oil or sluggish grease to prevent overshoot.

DATA-REDUCTION
the art or process of transforming masses of raw test or experimentally obtained data, usually gathered by instrumentation, into useful, ordered, or simplified intelligence.

DATA-REDUCTION, ON-LINE
the processing of information as rapidly as the information is received by the computing system.

DEBUG
to isolate and remove all malfunctions from a computer or all mistakes from a routine.

DECADE
a group or assembly of ten units, e.g., a decade counter counts to ten in one column; a decade resistor box inserts resistance quantities in multiples of powers of 10.

DECIMAL, CODED, BINARY
decimal notation in which the individual decimal digits are represented by some binary code, e.g., in the 8-4-2-1 coded decimal notation, the number twelve is represented as 0001 0010 for 1 and 2, respectively. Whereas in pure binary notation, it is represented as 1100. Other coded decimal notations are known as: 5-4-2-1, excess three, 2-4-2-1, etc.

DECODE

to ascertain the intended meaning of the individual characters or groups of characters in the pseudo-coded program.

DECODER

a device capable of ascertaining the significance or meaning of a group of signals and initiating a computer event based thereon; matrix.

DEFLECTION–SENSITIVITY

the quotient of the displacement of the electron beam at the place of impact by the change in deflecting field. It is usually expressed in millimeters per volt applied between the deflection electrodes, or in millimeters per gauss of the deflecting magnetic field.

DELAY–LINE, ELECTRIC

a transmission line of lumped or distributed capacitive and inductive elements in which the velocity of propagation of electromagnetic energy is small compared with the velocity of light. Storage is accomplished by re-circulation of wave patterns containing information, usually in binary form.

DELAY–LINE, MAGNETIC

a metallic medium along which the velocity of propagation of magnetic energy is small relative to the speed of light. Storage is accomplished by recirculation of wave patterns containing information, usually in binary form.

DELAY–LINE, MERCURY or QUARTZ

a sonic or acoustic delay-line in which mercury or quartz is used as the medium of sound transmission. See Delay-line, Sonic or Acoustic.

DELAY–LINE, SONIC or ACOUSTIC

a device capable of transmitting retarded sound pulses, transmission being accomplished by wave patterns of elastic deformation. Storage is accomplished by re-circulation of wave patterns containing information, usually in binary form.

DENSITY PACKING

the number of units of useful information contained within a given linear dimension, usually expressed in units per inch, e.g., the number of binary digit magnetic pulses stored on tape or drum per linear inch on a single track by a single head.

DESIGN, LOGICAL

(1) The planning of a computer or data-processing system prior to its detailed engineering design. 2) The synthesizing of a network of logical elements to perform a specified function. 3) The result of 1) and 2) above, frequently called the logic of the system, machine, or network.

DIAGRAM

a schematic representation of a sequence of subroutines designed to solve a problem; a coarser and less symbolic representation than a flow chart, frequently including descriptions in English words; a schematic or logical drawing showing the electrical circuit or logical arrangements within a component.

DIAGRAM, LOGICAL

in logical design, a diagram representing the logical elements and their interconnections without necessarily expressing construction or engineering details.

DIFFERENTIATOR

a device whose output function is proportional to a derivative of its input function with respect to one or more variables.

DIGIT

one of the *n* symbols of integral value ranging from 0 to *n-1* inclusive in a scale of numbering of base n, e.g., one of the ten decimal digits, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

DIGIT, BINARY

a whole number in the binary scale of notation; this digit may be only 0 (zero) or 1 (one). It may be equivalent to an "on" or "off" condition, a "yes" or a "no", etc.

DIGIT, DECIMAL, CODED

one of ten arbitrarily-selected patterns of ones and zeros used to represent the decimal digits.

DIGITAL

the quality of utilizing numbers in a given scale of notation to represent all the quantities that occur in a problem or a calculation.

DIGITIZE

to render an analog measurement of a physical variable into a numerical value, expressing the quantity in digital form.

DIGITS, CHECK

one or more redundant digits in a character or word, which depend upon the remaining digits in such a fashion that if a digit changes, the malfunction can be detected, e.g., a given digit may be zero if the sum of other digits in the word is odd, and this (check) digit may be one if the sum of other digits in the word is even.

DIGITS, EQUIVALENT BINARY

the number of binary digits required to express a number in another base with the same precision, e.g., approximately 3 1/3 times the number of decimal digits is required to express a decimal number in binary form. For the case of coded decimal notation, the number of binary digits required is 4 times the number of decimal digits.

DOWN–TIME

the period during which a computer is malfunctioning or not operating correctly due to machine failures; contrasted with available time, idle time or standby time.

DRUM, MAGNETIC

a rotating cylinder on whose magnetic-material coating information is stored in the form of magnetized dipoles, the orientation or polarity of which is used to store binary information.

DUMMY

an artificial address, instruction, or other unit of information inserted solely to fulfill prescribed conditions (such as word-length or block-length) without affecting operations.

DUMP, A. C.

the removal of all A. C. power, intentionally, accidentally or conditionally from a system or component. An A. C. dump usually results in the removal of all power.

DUMP, D. C.

the removal of all D.C. power, intentionally, accidentally, or conditionally, from a system or component.

DUMP, POWER

the removal of all power accidentally or intentionally.

ECCLES–JORDON (TRIGGER)

a direct coupled multivibrator circuit possessing two conditions of stable equilibrium. Also known as a flip-flop circuit or "toggle".

ECHO CHECKING

a system of assuring accuracy by reflecting the transmitted information back to the transmitter and comparing the reflected information with that which was transmitted.

EDIT

to rearrange information. Editing may involve the deletion of unwanted data, the selection of pertinent data, the insertion of invariant symbols such as page numbers and typewriter characters, and the application of standard processes such as zero-suppression.

ELECTRONIC

pertaining to the application of that branch of science which deals with the motion, emission and behavior of currents of free electrons, especially in vacuum, gas or phototubes and special conductors or semi-conductors. Contrasted with electric which pertains to the flow of large currents in wires only.

ELEMENT, LOGICAL

in a computer or data-processing system, the smallest building blocks which can be represented by operators in an appropriate system of symbolic logic. Typical logical elements are the and-gate and the flip-flop, which can be represented as operators in a suitable symbolic logic.

ELEVATION

the angular measurement in a vertical plane from a specific reference, usually the horizontal plane.

ENCODER

a network or system in which only one input is excited at a time and each input produces a combination of outputs. Sometimes called a matrix.

ERASE

to replace all the binary digits in a storage device by binary zeros. In a binary computer, *erasing* is equivalent to clearing, while in a coded decimal computer where the pulse code for decimal zero may contain binary ones, *clearing* leaves decimal zero while *erasing* leaves all-zero pulse codes.

ERROR

the amount of loss of precision in a quantity; the difference between an accurate quantity and its calculated approximation; *errors* occur in numerical methods; *mistakes* occur in programming, coding, data transcription, and operating; *malfunctions* occur in computers and are due to physical limitations on the properties of materials; the differential margin by which a controlled unit deviates from its target value.

ERROR, INHERITED

the error in the initial values; especially the error inherited from the previous steps in the step-by-step integration.

ERROR, ROUNDING

the error resulting from deleting the less significant digits of a quantity and applying some rule of correction to the part retained. A common round-off rule is to take the quantity to the nearest digit. Thus, pi, 3.14159265. . ., rounded to four decimals is 3.1416. Note; Alston S. Householder suggests the following terms: "initial errors", "generated errors", "propagated errors" and "residual errors". If x is the true value of the argument, and x^* the quantity used in computation, then, assuming one wishes f(x), $x-x^*$ is the initial error; $f(x)-f(x^*)$ is the propagated error. If f_a is the Taylor, or other, approximation utilized, then $f(x^*)-f_a(x^*)$ is the residual error. If f^* is the actual result then f_a-f^* is the generated error, and this is what builds up as a result of rounding.

ERROR, TRUNCATION

the error resulting from the use of only a finite number of terms of an infinite series, or from the approximation of operations in the infinitesimal calculus by operations in the calculus of finite differences.

EXCHANGE

to interchange the contents of two storage devices or locations.

EXTRACT

to remove from a set of items of information all those items that meet some arbitrary criterion; to replace the contents of specific parts of a quantity (as indicated by some other quantity called an extractor) by the contents of specific parts of a third quantity, e.g., if the number 01101 is stored, the machine can remove and act upon or according to the third digit, in this case a 1.

FACTOR, SCALE

one or more coefficients used to multiply or divide quantities in a problem in order to convert them so as to have them lie in a given range of magnitude, e.g., plus one to minus one.

FEED, CARD

a mechanism which moves cards serially into a machine.

FERROELECTRIC

a phenomenon exhibited by materials within which permanent electric dipoles exist and a residual displacement in the D–E plane occurs.

FERROMAGNETICS

in computer technology, the science that deals with the storage of information and the logical control of pulse sequences through the utilization of the magnetic polarization properties of materials to store binary information.

FIELD

a set of one or more characters (not necessarily all lying on the same word) which is treated as a whole; a set of one or more columns on a punched card consistently used to record similar information.

FIELD, CARD

a set of card columns fixed as to number and position into which the same unit of information is regularly entered.

FILE

a sequential set of items.

FIXED–POINT

a notation or system of arithmetic in which all numerical quantities are expressed by a predetermined number of digits with the point implicitly located at some pre-determined position; contrasted with floating-point.

FLIP–FLOP

a bi-stable device; a device capable of assuming two stable states; a bi-stable device which may assume a given stable state depending upon the pulse history of one or more input points and having one or more output points. The device is capable of storing a bit of information; controlling gates, etc.; a toggle. See Eccles Jordan.

FLOATING–POINT

a notation in which a number x is represented by a pair of numbers y and z (and two integers n and m which are understood parameters in any given representation) with y and z chosen so that $x = y \cdot nf^z$ where z is an integer, ordinarily either $m > |y| \gg m/n$, or $y = 0$ (where n is usually 2 or 10 and m is usually 1). The quantity y is called the fraction or mantissa; the integer z is called the exponent or characteristic, e.g. a decimal number 241,000,000 might be shown as 2.41, 8, since it is equal to 2.41×10^8.

FLOW–CHART

a graphical representation of a sequence of operations, using symbols to represent the operations such as compute, substitute, compare, jump, copy, read, write, etc.. A *flow chart* is a more detailed representation than a *diagram.*

FORCE

to intervene manually in a routine and cause the computer to execute a jump instruction.

FOUR–ADDRESS

see code, Multiple-address.

FUNCTION–TABLE

two or more sets of information so arranged that an entry in one set selects one or more entries in the remaining sets; a dictionary; a device constructed of hardware, or a subroutine, which can either (a) decode multiple inputs into a single output or (b) encode a single input into multiple outputs; a tabulation of the values of a function for a set of values of the variable.

FUNCTOR

a logical element which performs a specific function or provides a linkage between variables.

JOINT COMPUTER COMMITTEE

SENEWS

SCIENCE EDUCATION SUBCOMMITTEE NEWSLETTER

VOL. I NO. 2 September 1958

PURPOSE

SENEWS is a newsletter addressed to computer oriented members of IRE, ACM, AIEE, to help them promote interest and knowledge among high school age students; it provides a nationwide medium of communication pinpointed to this subject; it evolves from volunteer efforts of the JCC Science Education Subcommittee and relies on its readers for news material.

Write to: C. W. Farr, Chairman
JCC Science Education Subcommittee
M.I.T. Lincoln Laboratory
Lexington 73, Massachusetts

. . . or to your representative on the SENEWS Editorial Board: Richard W. Melville, IRE; George E. Forsythe, ACM; G. L. Hollander, AIEE.

DANGEROUS VOLTAGE

SENEWS bristles with stories of computer activity in high schools; but *do not* expect your local authorities to welcome you with open arms when you knock on the door and announce your intention to help them "discover" computers. Dr. L. C. VanAtta, a leader in the energetic and successful industry-education program in southern California, has written us.

> "A word of warning. It has been my experience that school systems tend to regard individual companies, specialized professional groups (petroleum engineers, aeronautical engineers, electrical engineers), and others as pressure groups unless their offers of assistance to the schools are on a very broad basis. Naturally the schools and community are interested in the total education of the student, rather than in biasing the student toward any particular specialization."

Dr. VanAtta went on to point out that innovations (no matter how good) represent a disruption to established curricula. Since Sputnik the well-meaning offers of assistance have multiplied. In Los Angeles the superintendent of schools has created a central committee for cooperation, and the school system has provided a full time Executive Secretary to coordinate the community efforts and "arrive at a broad program in which all specialized subjects have their appropriate emphasis."

Avoid burning up the circuit; measure the input impedance of the school system in your community before applying your driving voltage.

HAVE DESIGN, WILL BUILD—CHAPTER TWO

In the last issue we reported how David Ecklein, a high school junior from Cedar Falls, Iowa, was building a checker playing digital computer at home, using his own design, 3500 surplus tubes, and the help of his buddies in Tom Sawyer style. (Editor's Note: Last minute correction of number of tubes was not in time for all publications.)

David's project has been interrupted for a while by summer employment at IBM's research laboratory in Poughkeepsie, New York, where he will learn to program and operate one of the major computers. He is looking forward to talking with engineers about circuit designs, and learning about the well-known checker playing program of Dr. A. L. Samuel. Nobody who has met 17-year-old David will be greatly surprised if he makes some significant improvements in the techniques before the summer is out.

David Ecklein's opportunity to work in a major computer laboratory came about through his contact with the JCC Science Education Subcommittee. It is, in David's words, "beyond my fondest hopes. Not only will it provide experience but also a means to earn funds to carry out my project to conclusion."

Werner Buchholz

JUNIOR HIGH SCHOOL COMPUTER PROGRAMMERS

The thirteen-year-old in your home may not have learned to write the following coded program for the IBM 704 computer to solve (for 5 values each of a, b, and c) the equation $P=(a+b)(a^2+ab+b^2)(c+1)$. But don't be discouraged. We know of only one Junior High which has provided such instruction.

```
Symbol   Operation   Address, Tag, Decrement          Remarks
SBM      LXA         *5 HERE, 1               Load index register

         CIA         *A DATA+5,1                    (a+b)
         FAD         *B DATA+5,1
         STO         *TEMP

         LDQ          A DATA+5,1                     a^2
         FMP          A DATA+5,1
         STO          TEMP+1

         LDQ          A DATA+5,1                     ab
         FMP          B DATA+5,1
         STO          TEMP+2

         LDQ          B DATA+5,1                     b^2
         FMP          B DATA+5,1
         STO          TEMP+3

         CLA          TEMP+1                    (a^2+ab+b^2)
         FAD          TEMP+2
         FAD          TEMP+3
         STO          TEMP+4

         CLA         *C DATA+5,1                    (c+1)
         FAD         *INFO
         STO          TEMP+5

         LDQ          TEMP                (a+b)(a^2+ab+b^2)=(PROD.)
         FMP          TEMP+4
         STO          TEMP+6

         LDQ          TEMP+5                 (PROD.)(c+1)=(ANS.)
         FMP          TEMP+6
         STO         *X INFO+5,1
```

```
TIX        SBM+1, 1, 1        If contents of index register are greater
                              than decrement (1), decrease contents by
                              decrement (1) and transfer to SBM+1.
                              Otherwise proceed to next instruction
                              (HTR)

HTR        SBM                Halt and return to starting position.
```

*Note: 5 is in location 5 HERE; 5 values of "a" start at location A DATA, 5 b's start at B DATA, 5 c's start at C DATA: number 1 is in location INFO; TEMP is a temporary storage register; X INFO is storage location of final result.

Harley Tillitt of the U. S. Naval Ordnance Test Station, China Lake, California, conducted an educational experiment with 24 selected eighth grade students at Burroughs Junior High School in November 1957. The experiment involved lectures and demonstrations during 10 forty minute instruction periods. Computer programmers will recognize the conventions developed by SHARE, an informal programming organization of IBM 704 users.

"The success of the students in these experiments shows that it is feasible to introduce computer programming instruction below the college level," concluded Tillitt in the report submitted to SENEWS. The problem coded above is one of 14 problems assigned to eighth grade students during the experiment. The complete report came to SENEWS via George Forsythe. (Editor's Note: If this report is published in more detail, SENEWS will announce it in a later issue.)

SADSAC II

Samson Additive Digital Sequential Automatic Computer, SADSAC II, carried its designer-builder, Peter Samson to first place in the Massachusetts State Science Fair, and fourth place in the 1958 National Science Fair. Peter lives in Lowell, Massachusetts, and will enter M.I.T. this autumn with a four year National Merit Award scholarship. He visited Lincoln Laboratory recently on invitation from the Science Education Subcommittee—and when he left it was *we* who were inspired.

First we reviewed specifications. SADSAC II is a relay computer, designed to add, subtract, and complement four bit binary numbers; input is from two punched paper tape readers, and programming provides for jumping from one tape to the other; output is a solenoid actuated typewriter (actually a home rigged 1898 Oliver).

But then we explored motivation, and the "engineering economics" of student computers. Peter's early science interest found roots in his father's electrical (hi-fi, etc.) equipment. High school algebra whetted his mathematics appetite. Attempting to build a machine to play tic tac toe led him to design his own arithmetic circuitry, and the home grown SADSAC I computer was the inevitable result. After winning local honors with SADSAC I he went on to national fame with its successor.

Peter did not have help from a computer expert advisor; like most student builders of computers he dug the know-how out for himself, "and used common sense," he hastens to add. He regarded pertinent literature as scarce but good; the hardware situation he found pretty sad. We asked why he used relays instead of tubes or transistors. He pointed out that relays are cheap and are AC energized at a few standardized voltages; tubes and transistors involve not only more money, but more complex circuitry and power supplies. He used discarded pinball machine components, and reliability was a nightmare.

"Building SADSAC III this summer, Peter?"

"No. I want to get in some study that I can't find time for during the school year."

BOOK AND FILM

LOCAL ACTION KIT, prepared by The President's Committee on Scientists and Engineers, Washington 25, D. C.

This kit is designed to:

Tell how to bring together community organizations with the same basic objective into a single program.

Promote self-evaluation and the recording of successful techniques and experiences useful to other communities throughout the nation.

Establish a pattern of nationwide effort without sacrificing the incentives and benefits of creative local action.

Here is what the kit contains:

A Guidebook for Local Action, suggesting general techniques adaptable to the needs and resources of your own community.

Examples of tested projects in other communities—relating how they began; how they are being carried out; their scope, financing and results.

Reference materials such as information on scholarships, improvement of science curricula, youth activities, etc.

A bibliography of selected materials, visual aids, organizations and publications helpful in your community program.

This kit was prepared for use in local programs for the improvement of science and mathematics education (not limited to computer education). It is available without charge to *groups*, not (for budgetary reasons) to individuals.

Interested individuals can get free copies of Local Action, the monthly news bulletin.

Editor's Note: Don't be misled by the price; this is good stuff.

PAPAC-00, A DO-IT-YOURSELF PAPER COMPUTER

In less than an hour you can build the simplified digital computer shown in Figure 1, using only a pair of scissors, three dozen common pins, and the parts shown in Figures 1 and 2. This computer was developed from the model demonstrated in the Concord, Massachusetts, High School lectures on computers reported in SENEWS Volume I, Number 1.

From the discussion below, the computer expert will recognize that "*PAPAC double zero*" contains most of the units of a large-scale computer, but in simplified form. The *control unit* includes a counter and a system for controlling the parts of the computer according to the instruction being performed (in this model a simple fixed instruction is used; a large computer can draw from several instructions obtained from storage). The *storage unit* includes registers, bus, and selection switch; register contents are changed by hand rather than by the computer. The *arithmetic unit* can add. *Input and output units* have been eliminated by allowing the operator to deal with the insides of the computer directly rather than by way of complicated equipment. Proprietary rights are held by the author.

In *operation,* PAPAC-00 follows the same fixed instruction over and over again. This instruction is: "Read the number out of the currently-selected storage register and add it to the adder, then get ready to use the next storage register for the next time." The "Counter" keeps track of which storage register to use next; since there are only two registers, numbered "0" and "1," the counter alternates between them. The "Switch" is controlled by the counter and allows only the selected register to be operated. Each "storage register" contains only a single binary "cell"; when the register is operated, the cell is forced against the "Bus" if the cell is set to "1." If a "1" has been read out in this way, the bus actuates the "Adder," preparing it to add the "1." If the cell is set to "0," the bus and adder are not operated, and "0" is added to the adder. Binary sums are as follows: 0+0=0, 0+1=1, 1+0=1, 1+1=10. The adder forms these sums correctly except that in the last case it forms a sum of "0" because it can handle only one digit. The "Control," pushed back and forth by hand, performs this fixed instruction by operating the counter and switch, and by returning the bus to its "0" position (if it had read out a "1") causing the sum to be formed in the adder.

To *assemble* PAPAC-00, Figure 1 should be used as the base, and the shapes of Figure 2 should be fitted over it by following these steps:

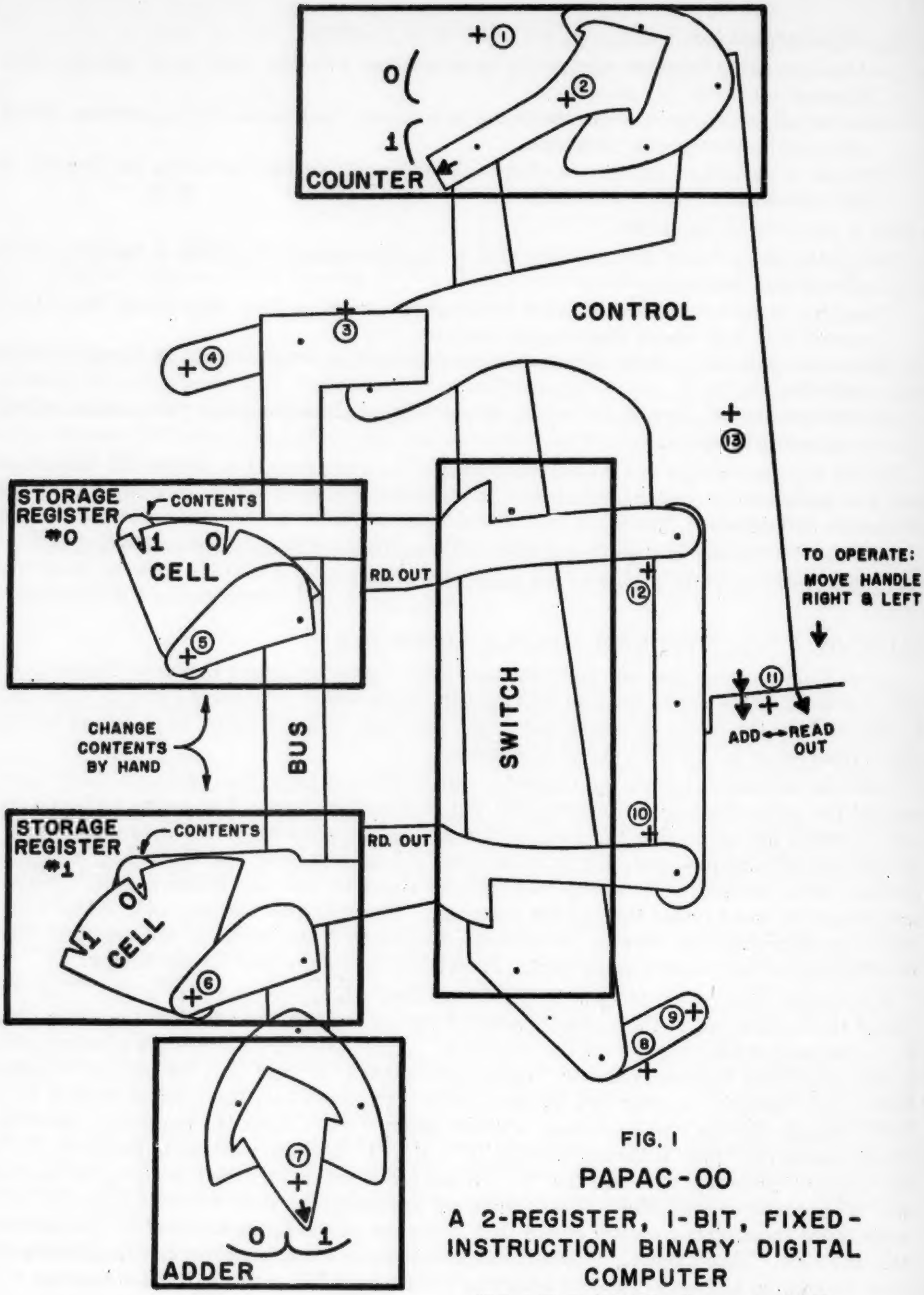


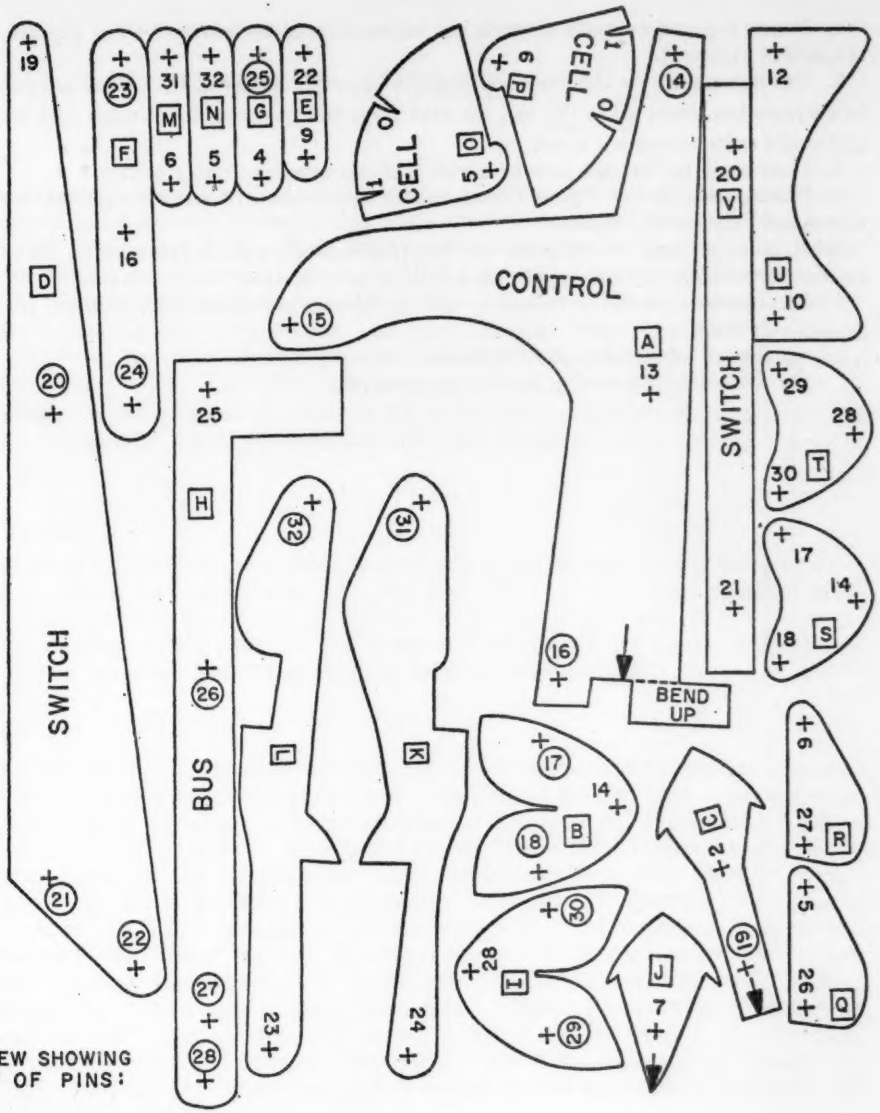
FIG. 1

PAPAC-00

A 2-REGISTER, 1-BIT, FIXED-INSTRUCTION BINARY DIGITAL COMPUTER

(ROLLIN P. MAYER, JULY 14, 1958)

A-87379



NOTE: EDGE VIEW SHOWING ARRANGEMENT OF PINS:

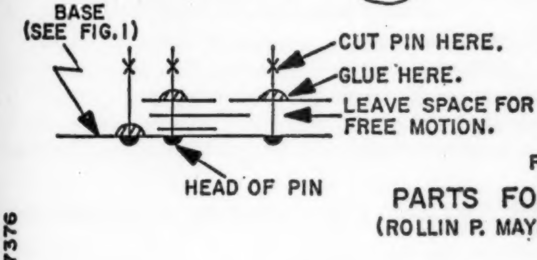


FIG. 2

PARTS FOR PAPAC-OO

(ROLLIN P. MAYER, JULY 14,1958)

A-87376

1. Punch a pinhole exactly through the intersection of each cross (+) in Figures 1 and 2 (but not the dots in Figure 1).

2. Cut out exactly on the lines, the parts in Figure 2, in any order. They are marked with a letter in a square box, from $\boxed{A}$ to $\boxed{V}$, and the next steps will be easier if you place each piece on the table in alphabetic order as you cut it out.

3. Place a pin up through each hole with a circled number (from *1* to *32*).

4. Taking each part of Figure 2 in alphabetic order, place its *un*circled number holes down over the correspondingly numbered pins.

5. In first operating the computer you may find that some parts jam because the upper piece is down too far on the pins: pry such pieces up a little to provide space for free motion.

6. The construction can be refined by cutting the pins and gluing the uppermost part to the remaining length. Caution:

   (a) Don't cut the stop pins too short.

   (b) Glue only *one* moving part to the same pin.

Rollin P. Mayer

NEWS & NOTICES

Cleveland Chapter:

Formal efforts to organize a Cleveland Chapter began in late 1957; at two subsequent organizational meetings the speakers included Melvin Conway of Case Institute of Technology ("Practical Compilers") and John W. Carr, III, President of ACM. The Chapter held its first Annual Meeting on May 20 of this year. Dr. Tibor Fabian, visiting professor from the UCLA Business School to the Operations Research Group of Case Institute of Technology, presented the theme, "How to Devise a Business Game."

Officers elected for the coming year are as follows: Chairman, Frederick Way, III of Case Institute of Technology; Vice Chairman, Raymond Hitti of Standard Oil of Ohio; and Secretary-Treasurer, Sally Dennis of IBM Corporation.

The Executive Council includes: Lynn V. Albers of Lewis Laboratory, NACA; Winston Riley, III of IBM Corporation; Bernard N. Riskin of Chesapeake and Ohio Railway; and L. Richard Turner of Lewis Laboratory, NACA.

Delaware-Valley Chapter:

This new chapter of seventy members was officially approved at the council meeting held as part of the national ACM meeting at the University of Illinois in June, 1958. Membership is expected to increase rapidly since there are more than 200 ACM members in the Philadelphia area.

At the first official meeting on July 23 at the University of Pennsylvania, Dr. Grace M. Hopper of Remington Rand Univac spoke on "Developments in Automatic Programming."

Los Angeles Chapter:

A one-day technical symposium, to stimulate the exchange of information on decision-making applications of computers and to provide an opportunity for newer members of the profession to participate in a full-fledged technical meeting, was sponsored by the Los Angeles Chapter of ACM at UCLA, on Thursday, August 21, 1958. The program included ten technical papers: "The SPADE Project for the Solution of Partial Differential Equations" by R. Finkel of Ramo-Wooldridge; "Automatic Memory Storage Allocation" by J. J. Peterka and S. H. Wong both of System Development Corporation; "A Dynamic Storage Allocation System" by J. B. Rogers and P. Weaver, both of System Development Corporation; "Optimization of Rocket Nozzle Design" by D. L. Bussard of Ramo-Wooldridge; "Printed Circuit Card Wire Layout Program" by E. A. Riordan and K. V. Tooker, both of IBM (Military Products); "Scaling, the Rhyme and Reason" by A. Hammond of Ramo-Wooldridge; "Chemical Equilibrium Program" by R. Paul and G. Reitz, both of Marquardt Aircraft Company; "The Numerical Control of a Drivmatic Riveter" by R. Notestine of Lockheed Aircraft Corporation; "The SDC Input-Output System" by R. A. Brouse of System Development Corporation; and "Decision Making in Industrial Situations" by M. Asimow of UCLA's Engineering School.

Gerhard Reitz of Marquardt Aircraft and Donn Combelic of Ramo-Wooldridge, served as chairmen of the morning and afternoon sessions respectively. George W. Brown, Director of the Western Data Processing Center, was the guest speaker at the luncheon served between sessions. Preprinted summaries of the papers were distributed to all registrants at the door. All arrangements for this symposium were handled by a committee under the chairmanship of J. D. Madden of System Development Corporation.

UNIVERSITY ACTIVITIES AND EDUCATION PROGRAMS

- Professor C. C. Gotlieb of the University of Toronto has provided the following list of university installations in Canada, dated July 4.

University	*City*	*Machine*	*Remarks*	*Persons for Reference*
Toronto	Toronto, Ont.	IBM 650 since May, 1958 (Ferranti MkI from 1952–58)	Extras: include 4 magnetic tapes, 3 index accumulators, floating point.	Prof. C. C. Gotlieb, Physics
Manitoba	Winnipeg, Man.	Bendix G-15D		Prof. Mendlessohn (N.) Mathematics
Saskatchewan	Saskatoon, Sask.	LGP-30		Prof. N. Shklov, Mathematics
Alberta	Edmonton, Alta.	LGP-30		Prof. M. Wyman, Mathematics
British Columbia	Vancouver, B. C.	ALWAC III-E		Prof. T. Hull, Mathematics
Ottawa	Ottawa, Ont.	IBM 650	On order	Prof. A. Smith, Mathematics
McGill	Montreal, Que.	Datatron 205	Recommended	P. R. Wallace, Physics
Laval	Quebec, Que.	IBM 650	Recommended	Prof. Boisvert, Physics

• The University of Durham's new Computing Laboratory in Newcastle upon Tyne is one of the first in England to be equipped with a Ferranti Pegasus Computer. The cost of FERDINAND (FERranti DIgital Numerical Analyzer, Newcastle and Durham) was met, in part, by assistance from the University Grants Committee and local industry.

Machine time is shared by University departments and outside users on a time-hire basis. Several programming courses, including two for under-graduates, have been run since the Laboratory's opening in January, 1958.

• Professor John W. Hamblen of the Oklahoma A and M College Computing Center has accepted a position as Director of the Computing Center and Associate Professor of Statistics at the University of Kentucky, effective September 1.

• Dr. Michael Barnett of the Naval Research Laboratory at the University of Wisconsin has accepted a position with the Solid State and Molecular Physics Group of the Department of Physics at MIT.

• The program of the University of Michigan Engineering Summer Conference on Numerical Analysis, Artificial Intelligence, and related topics during June, 1958, included talks by guest lecturers A. A. Dorodnicyn, of the Soviet Academy of Sciences, and Professors L. N. Korolev and V. S. Burtsev of the Soviet Union. Their lectures were on "Method of the Integral Relations for the Numerical Solution of Partial Differential Equations," "Some Methods of Automatic Coding for BESM and STRELA Computers," and "Accelerating Multiplication and Division in High-Speed Digital Computers," respectively. The fourth member of the Soviet delegation was Professor I. S. Mukhin.

• From *Remington Rand News*, June, 1958: UNIVAC installations at a number of universities across the nation are making important contributions to the advancement of science, the training of technical personnel, and the development of new procedures for the benefit of business and industry. Following are some of the institutions of higher learning which have established computing centers built around UNIVAC I: Harvard University, the University of Pennsylvania, Case Institute (Cleveland), and the University of Chicago. UNIVAC Scientific models have been installed at Georgia Tech., Southern Methodist University and, this month, at the University of Minnesota. UNIVAC 120 and 60 computer users include the University of Denver, University of Syracuse, Mohawk Valley Technical Institute in New York State, and the Baltimore County Board of Education. Plans are going forward for the establishment of UNIVAC computing centers at a number of other universities during the months ahead.

• The University of Michigan IBM 650 installation is working on a new compiler planned to be in operation by mid-September, 1958. The language is essentially that of the IT Compiler, with the following new features: (1) Expanded input-output facilities, including the use of alphanumeric information and allowing variable names to be in alphanumeric form during execution of the object program; (2) More flexible variable designation, including greater freedom in the use of arrays; (3) Conventional hierarchy of parentheses and operations.

The decomposition of the input statement will result in a sequence of three-address machine-independent instructions which will then be translated into an optimized IBM 650 machine language program. The translation of the three-address program and the optimization will be an integral part of the compiler, which will need both special character devices, floating point arithmetic, and index accumulators. Inquiries may be addressed to Mr. Bruce Arden, Computer Laboratory, 110 Rackham Building, University of Michigan, Ann Arbor, Michigan.

- The Computing Center of the Division of Applied Mathematics of Brown University, Providence, Rhode Island, has not yet been included in our list of IBM 650 installations. The installation is under the direction of Professor Walter F. Freiberger.
- Data Reduction Centers at the University of Wisconsin and Ohio State University will be established under the terms of a Federal contract accepted by both universities early in May. The Centers will relate and interpret scientific observations made in Antarctica during the International Geophysical Year. The two schools were named by the National Academy of Sciences' IGY Committee; funds amounting to $86,820 will be furnished by the National Science Foundation. Work on each campus will cover gravity, magnetic, and seismic observations made by American parties in Antarctica and similar observations to be carried out by U. S. scientists during 1958–59. Professor George P. Woollard, who directs the University of Wisconsin's geophysics studies, heads the new office at his school. The agency at Ohio is under the direction of Dr. Richard Goldthwait. (Courtesy *Datamation*)

- ALWAC Computer Division has announced three training courses for the month of September, 1958 to be held at its main plant in Hawthorne, California: Elementary Digital Computer Programming—P-1, September 8–19; Introduction to Digital Computers (Executive Seminar)—EX-1, September 24–26; and Digital Computer Maintenance and Field Service—FS-1, September 22–October 24. Executive seminars and programming courses are also held periodically at ALWAC regional offices in San Francisco, Cleveland, Washington, D. C., and New York.

For information concerning any of the above courses, contact the Regional ALWAC Computer Division Offices or Mr. Jack Thompson, Director of Education at the Hawthorne facility, 13040 South Cerese Avenue, Hawthorne, California.

COOPERATIVE PROGRAMMING GROUPS

SHARE:

On June 23 and 24 a meeting was held at North American Aviation, Inc., in Los Angeles, California, between representatives of IBM and several SHARE member installations for the purpose of informing prospective 709 FORTRAN users of current progress and status, evaluating this progress, and formulating specifications for future revisions. Two subcommittees were appointed to evaluate the current FORTRAN systems for the 704 and the 709. These subcommittees will study proposed revisions and make recommendations to the Applied Programming Group of IBM concerning specifications for subsequent FORTRAN systems for future compilers.

It was announced that an 8K version of the 709 FORTRAN compiler will be available in late August, 1958. The 32K version will be made available "shortly thereafter." Operation of the 709 version in the "Monitor Mode" (i.e. tape-to-tape, multiple jobs) will be facilitated but not actually accomplished.

UNIVAC Users Conference:

The next meeting of the UNIVAC Users Conference will be held on October 20 and 21, 1958 at Boston, Massachusetts. The John Hancock Mutual Life Insurance Company will act as host. Included in the meeting agenda are a visit to a UNIVAC I installation and a visit to the first UNIVAC II placed in customer service.

Eastern Council of DUO:

A combined technical and business meeting of Eastern DUO was held at Arthur D. Little, Inc., Acorn Park Laboratory in Cambridge, Massachusetts on July 18, 1958. Representatives from twenty-five Datatron computer installations were in attendance. The technical topics discussed included Integrated Personal Services Accounting, Point Set Compiler, Naval Shipyard Data Processing, Datatron Logic and Circuitry and Machine Learning Techniques. (*Dynamic Programming*)

The DUO members agreed to use the Burroughs Technical Bulletin format as a model in preparing new programs for distribution through the Users group. Burroughs will assume the task of printing and publishing these programs. It will also distribute abstracts of all programs to the Users, who then can request those needed.

NEWS ITEMS

- The ElectroData Division of Burroughs Corporation has announced a commercially available perforated-paper-tape reader which will operate at 1,000 characters per second. Operating photoelectrically, the reader will stop on a single stop character in a little more than one-thousandth of a second and will read the next character within five milliseconds of the next read operation. The reader can accommodate any standard-width tape, from five level to eight level. Available with the reader are reels which are equipped with an automatic brake and servo shut-off system to prevent tape damage and loss of data in the event of power failure or other malfunction. The reader is now in use as an input unit with the Burroughs 220 electronic data processing system. Delivery of the photoreader is quoted as four months after receipt of order.
- The Philco Corporation has announced performance data on the Transac S-2000. The computer is completely transistorized with a 65,536 word storage possible. Word length is 48 binary digits. The computer consumes seven to ten kilowatts of electricity and plugs into a conventional 110 volt, 60 cycle outlet. The computer performs computations on numbers equivalent to 15 decimal digits without programming double precision. The tape transports read and write 90,000 alphanumeric characters per second. The tapes include 16 channels with 12 information bits, two for parity and two for timing. Tape speed is 150 inches per second. An off-line printer is available with the S-2000 which utilizes a 1,024 character buffer and has a capability of 900 lines per minute.
- The Terrain Data Translator (TDT), a precision graphic-to-digital translating machine, was recently announced by Benson-Lehner Corporation of Los Angeles, California. The TDT consists of a first-order precision reading unit and an electronic counter. Two-axis output can be in the form of punched paper tape, punched cards or hard copy (remote control typewriter). This new device is expected to effect appreciable savings in the nation's highway construction program where it can be used to convert data from aerial photographs to digital form for input to computers to solve problems in cut and fill operations.
- Construction of one of the first digital computers to use new parametrons in its arithmetic and control units was completed in a Tokyo University Department of Physics laboratory and the computer is now in operation, according to Dr. Hidetoshi Takahashi, head of the laboratory.

The main feature of this computer is that it uses no transistors, diodes, or vacuum tubes in its arithmetic and control units. Instead, the model uses 3,800 parametrons, a discovery of a 27-year old Tokyo University research assistant, Eiichi Goto. Dr. Takahashi claims that a parametron computer can be built for about one-half the cost of a conventional vacuum tube circuit computer, tolerance is less rigid and possibility of breakdown is remote, and power consumption is relatively low. The main drawback in the parametron system is its slow speed, between 1/10 and 1/5 of vacuum tube speed. The clock cycle in the pilot model is 100 micro-seconds, but the next model will be three times as fast, and improvement up to ten times is forecast for future models. The high frequency power supply which is required is also considered a disadvantage.

Plans now under discussion call for a prototype commercial model of the parametron computer to be built at Tokyo University by a subsidiary of Fuji Electric Company, one of Japan's largest integrated electrical manufacturers and another to be built by Yurin Electric Company, which is presently operating Japan's only computer service. (Courtesy *Datamation*)

• Packard Bell Computer Corporation, which recently moved into a new home at 1905 Armacost Avenue, Los Angeles 25, California, announced that Air Force Missile Development at Holloman Air Force Base has ordered a Digital/Analog Function Table System. This new DAFT System will involve an arbitrary function generator, elements of a TRICE Computing System and voltage-to-digital, digital-to-voltage conversion (a multiverter). Basically the system will generate arbitrary functions digitally and then translate them to a voltage proportional to the digital numbers. It will be possible to have voltage input into the system so that two-way communication with an analog computer is possible.

• Philco Corporation and Leeds and Northrup Company have announced a shared-cost program to develop, design and build a digital computer for applications in industrial process control and in scientific and engineering computation including data processing. This development will bring together the experience of Leeds and Northrup in the process control field and Philco's knowledge of transistorized digital computers. It is expected that the computer will have application for control systems for chemical plants, petroleum refineries, electric power plants, atomic energy processes, steel mills and metal working plants. Also, the computer will be available for the processing of data and engineering computations, for real time control for military systems and for other commercial uses.

• A mobile computing system, including a high-speed special purpose electronic computer with display output which consists of a projected and numerical presentation, was recently designed and built by Remington Rand and accepted by the Air Force for the Air Development Center at Rome, New York. The new equipment, which is housed in two 35-foot drop frame semi-trailers and powered by two diesel motor generators, provides a technique for controlling electronic warfare devices and represents a future requirement of modern warfare (i.e., the use of automatic data handling to provide optimum decisions within the required time limit).

• During the past month, Litton Industries and the Canadian Air Research and Development Establishment have both announced acceptance of their respective ALWAC III-E Magnetic Tape systems. C.A.R.D.E. marks the fifth tape installation since the Personnel Research Board of the Adjutant General's Office received the first tape buffer and two units in March of this year. The very high speed and core-buffered ALWAC Magnetic Tape Systems are now in full production and are being delivered according to normal schedule.

• New ALWAC III-E Data Processing Systems installed and accepted during the month of June include the Technical Research Group in New York City and the Strategic Air Command Headquarters at Offut Field in Omaha, Nebraska. It is planned that these installations, now including the high speed punched paper tape systems and the large 8,192 word magnetic drums, will ultimately be expanded to full scale installations, including both punched card buffers and magnetic tape complexes.

• An IBM 705, installed at Mare Island Naval Shipyard in Vallejo, California, is the first computer to operate in any U. S. naval shipyard. Rear Adm. Martin J. Lawrence, Shipyard Commander, states that the system will be used to assist shipyard officials in nearly every phase of the yard's complex $80,000,000-a-year operation and to coordinate and speed construction of the new Polaris missile-launching submarine being constructed at the yard.

• Consolidated Electrodynamics Corporation, Pasadena, California will soon begin to deliver new advanced airborne data processing instrumentation to Douglas Aircraft Company for the flight-test program for Douglas DC-8 jet airliners. The total contract, for development and manufacture of the digital recording and magnetic tape equipment exceeds $2,250,000.

• The May/June issue of *Datamation* magazine described the design of a new Telemeter Magnetics data translator as a unit "based on the building block concept whereby custom installations to suit individual needs can be constructed from standard modules. These data translators provide compatibility between computers of different designs via almost any conceivable combination of punched cards, paper tape or magnetic tape. Reliability and accuracy are said to be high owing to the use of solid state components and internal automatic checking codes."

• System designers are now provided with complete sets of building blocks for digital magnetic tape applications by completely transistorized read/write amplifiers manufactured by Decisional Control

Associates of Santa Ana, California. The circuits may be incorporated into any type of digital system, and a typical read/write system for eight channels is contained in a single module occupying 3½ inches of panel space. (Courtesy *Datamation*)

- The first commercial installation of an IBM 650 RAMAC will be for Brown and Bigelow in St. Paul, Minnesota. This high speed data processing and random access memory machine, valued at $546,000, will be utilized in the company's direct mail, calendar, playing card, greeting card, and novelty business.

- By adapting its assessing, auditing and treasury department procedures to the use of computing machines, the city of Boston, Massachusetts, is expected to save several million dollars annually. Although Boston only recently purchased its first UNIVAC 60 computer, future plans include computing of real estate taxes on approximately 125,000 separate properties, poll taxes, and all other billing and purchasing done by the city.

- With the installation of a UNIVAC 60 computer, Broadway Bank and Trust Company of Paterson, New Jersey, will be the first of the so-called small banks (assets between $25-million and $200-million) to handle its entire accounting system with electronic equipment. Equipped with a system of self-checking digits which guarantees the correct posting of correct amounts to proper accounts, the computer will handle savings, installment loans, mortgages, income and expense distribution, liability ledger and general ledger accounts, and accounting on commercial and special checking accounts. It is anticipated that the one-time cost of installation will be recovered through savings within a few months.

- Model-O of the APT algebraic compiler is now operative on the 1103A at the Applied Physics Laboratory at Johns Hopkins University. It has been used successfully on several problems, including one of over 2000 instructions. Model-1, which frees the entire core for compilation, is nearing completion. APT is also being coded for ORDVAC at the Aberdeen Proving Ground. Those interested in coding APT for their machine may contact Dr. Robert Rich at Johns Hopkins University for details of the system.

- A new instrument capable of 2000 completely independent conversions per second and designed to operate either as a digital voltmeter or as an analog-to-digital converter, has recently been developed by ADAGE, Inc., in Cambridge, Massachusetts. The converter is completely transistorized to eliminate noise and heat and to reduce maintenance to a minimum. (Courtesy *Datamation*)

- In connection with the General Electric Company Lamp Division's research programs to determine how light bulbs burn out and to investigate new forms of light sources, a Bendix G-15 computer is being used to solve radiation problems involving temperature gradients in wire filaments and filament vibration.

- The Tudor Engineering Company of San Francisco and Palmer and Baker Engineers, Incorporated of Mobile, Alabama, are using digital Bendix G-15 computers to determine future traffic load and control in addition to highway location. Origin and destination data is fit into the computer to determine location of future roads.

In another problem computers are used to determine the highway load potential of various types of soils along future routes. The Highway Research Board of the National Academy of Sciences is directing a $22-million two-year test program scheduled to begin in September. The cost of the program will be shared by all the states, the Federal government, and the automobile and petroleum industry. Most of the $1-million worth of electrical equipment to be used is new and unique and was developed by the industry specifically for the test program. Computers will be utilized for the test data processing; a Bendix G-15 has been installed at the test site at Ottawa, Illinois, to process and analyze data from the more than 8,000 measuring devices on, in and under the test pavements.

- A. F. R. Brown's language translation program, described at the ACM meeting in Houston last year (Journal of the ACM, v. 6, No. 1, Jan. 1958, p. 1) has been coded for the 704. In a recent trial using a 600-word dictionary, fifty sentences of scientific French were successfully translated to English. The heart of the program is a macro- or pseudo-code used in the dictionary to express linguistic manipulations of the translation process. Dr. Brown has stated that the master program may be adapted to any language pair with relatively minor changes. The work has been conducted at Georgetown University, using the National Bureau of Standards' 704. A Russian-to-English translation program developed separately at Georgetown is scheduled to begin "dress rehearsals" in early summer. Both projects have been aided by a National Science Foundation grant.

• Lee Amaya, formerly of the California Division of Lockheed Aircraft at Burbank, has been transferred to Lockheed's Missile Systems Division. For this division he will be working in their Van Nuys facility until late 1958, when he will transfer to the Sunnyvale plant to take charge of the 709 installation.

• Reid Anderson, formerly Director of Physical Research at National Cash Register Company, Dayton, Ohio, has joined Stanford Research Institute as Manager of their Computer Laboratory. Mr. Anderson will plan and direct programs in the research and development of computer techniques, switching theory, high speed digital circuits, magnetic logic circuits, and special purpose computing devices. He will be responsible for three research groups: circuit systems, special devices, and computer development.

• Recent appointments at the ALWAC Computer Division, El-Tronics, Inc., include: Mr. Fred Garcelon, formerly Director of Marketing at CBS Hytron, Inc., who was named Executive Vice President of El-Tronics, Inc., and Mr. Andrew T. Fischer, formerly Marketing Director at ALWAC Computer Division, who was named Assistant to the President of El-Tronics, Inc. Messrs. Larry O. Seerden and John W. Busby, formerly Los Angeles and New York Sales Managers, respectively, have been appointed as Western and Eastern Regional Marketing Directors, respectively.

Six new additions to the ALWAC Marketing Department staff include: Thomas V. Cooper, Jr., formerly Regional Sales Manager at Remington Rand Univac; Jack W. Blaha, formerly of Remington Rand; Donald W. Spreen, formerly with IBM; Cloyd Kerr, formerly Manager of Tab Services at Consolidated Electrodynamics Corporation; George Grinnell, formerly with Remington Rand; and David Z. Polack, formerly with Remington Rand at Carborundum Company's UNIVAC installation. Messrs. Spreen and Cooper will be assigned to the New York office. Mr. Blaha will be headquartered in Cleveland and will work in the Midwest area. Mr. Grinnell will be assigned to the San Francisco office, and Messrs. Kerr and Polack will headquarter in the Los Angeles District Office.

• George Eisler, formerly in charge of systems formulation for National Cash Register, Electronics Division, has formed his own data processing systems consulting firm in Los Angeles which will be called Eisler Associates.

• Remington Rand recently announced the publication of six new brochures. "60-second Management Interview" which describes the methods used by a railroad to accomplish complete stores control using the UNIVAC 60; "Univac Math-Matic," which explains a new coding shorthand system which makes possible many mathematical computer applications never before practical and eliminates the need to write out detailed step-by-step coded computer instructions for UNIVAC I and UNIVAC II; a manual which details "Omniflex," an integrated UNIVAC II service routine which performs various functions including copying, correcting, searching, sampling, verifying, comparing, and analyzing; Report Number U-1419 describing the application of the UNIVAC 120 to earthwork computations (cross section plus and minus elevations, vertical curve correction, grade elevations, and center line cut/fill) in highway and engineering mathematics; and "Graphical Presentation with the UNIVAC 120 Punched-Card Electronic Computer" (Manual Number U-1471), describing techniques for converting large amounts of data into graphical form. A code of uniform "Programming Conventions for UNIVAC II" systems (Manual Number U-1433) has also been published. Adoption of the conventions by UNIVAC II users will make possible smooth exchange of standard programming routines among the users. The conventions include standard formats for data and instruction files and standard operating options for UNIVAC II routines. Each of these publications can be obtained at any branch office of Remington Rand or by writing Remington Rand, Division of Sperry Rand Corporation, 315 Fourth Avenue, New York 10, New York.

• A portable desk-size analog computer, Model MC-400, is fully described in a new brochure published by Mid-Century Instrumatic Corporation, 611 Broadway, New York 12, New York. The unit incorporates 16 scale-factor potentiometers, a precision servo multiplier, true overload alarm, monitor and control panel, vacuum-tube voltmeter, complete programming patchboard and three-way amplifiers.

• The Benson-Lehner Corporation of Los Angeles has announced the formation of a Military Products Division which will conduct research, development, and production of military electronic equipment. Technical areas of interest will include digital circuits and techniques, servomechanisms, analog computers, displays and electromechanical and optical-mechanical design. The new division will be managed by Mr. Harold J. Rounds, Jr., Vice President and General Manager of the Benson-Lehner Corporation. Mr. Joseph M. Cahn will be Chief Engineer.

• The mathematics department of General Kinetics Incorporated has moved to 542 23rd Street, South in Arlington, Virginia, nearby the main plant. The move is a part of the expansion of GKI's facilities.

• "Six Lectures on Management Sciences" were held in Washington, D. C. in May and June under the leadership of The Institute of Management Sciences and under the joint sponsorship of various organizations including ACM. The theme of the series was the application of scientific techniques to the making of management decisions. The speakers included: William W. Cooper, "Mathematics, Management Science, and Management"; Glen D. Camp, "Productions-Inventory-Distribution Operations as Special Cases of Queueing Operations"; Martin Tolcott, "Systems Approach in Human Engineering"; Howard Raiffa, "Game Theory and Decision Theory"; and Andrew Vaszonyi, "Application of Scientific Programming."

• Officers for the 1959 Western Joint Computer Conference at San Francisco have been appointed as follows: Conference Chairman, R. R. Johnson of General Electric (G.E.); Vice Chairman, R. W. Melville of Stanford Research Institute (SRI); Secretary-Treasurer, H. G. Asmus of G.E.; Technical Program, M. L. Lesser of IBM; Exhibits, H. K. Farrar of Pacific Telephone and Telegraph; Registration, R. M. Bennett, Jr., of IBM; Women's Activities, J. L. Teasdale of G.E.; Local Arrangements, R. C. Douthitt of Remington Rand; Publications, B. J. Bennet of IBM; Trips, K. F. Tiede of University of California Radiation Laboratory (UCRL); Publicity, G. A. Barnard of SRI; Mailing, E. T. Lincoln of SRI; and Printing, L. D. Krider of UCRL.

• A "Special Technical Conference on Non-Linear Magnetics and Magnetic Amplifiers" was held in Los Angeles, California, August 6, 7, 8, 1958. Two sessions were devoted to computer magnetics. Among the topics discussed were: "The State of The Art"; "New Very High Speed All Transistor Magnetic Core Memory"; "The Twistor—Its Use in A Memory Array"; "Logical Circuits Using Magnetic Cores and Capacitors"; "Magnetic Integrating Amplifier"; "Memory Element Fabrication by High Vacuum Thermal Evaporation Techniques"; "Magnetic Networks for Performing Logic"; "Ferro-resonant Switching Circuits"; "Thin Film Magnetization Reversal Studies"; "The Use of Magnetic Amplifiers at High Frequencies"; and "A High-Speed Logic System using Magnetic Elements and Connecting Wire Only." Informal seminar sessions were conducted on the following subjects: "Theory and Design"; "Applications"; and "Computer Magnetics."

• At the July, 1958 meeting of the Digital Computers Association, Mr. Milton Rosenberg of Telemeter Magnetics spoke on "Advanced High Speed Storage Techniques." Mr. Rosenberg stated that within the next year ferrite core memories will be available which have a moderate size (under 10,000 words) and a cycle time of 1 to 2 micro-seconds. The new memories, called "word selection" memories, will be approximately 50 per cent more expensive per word than existing memories. Memory words will be selected directly without benefit of coincidence in the array. The extra cost of the unit is due to a large decoding power switch.

Also discussed was a new memory element called the "Twistor," developed by A. H. Bobeck of Bell Telephone Laboratories. Storage depends on the principle of strain sensitivity (magnetostriction) in magnetic materials. The coincidence of a circular and a longitudinal magnetic field inserts information into a wire in the form of a polarized helical magnetization. The advantage of this type of system are a greater temperature range than present ferrite core memories and the possibility of applying weaving techniques in the assembly process. A 4000-word, 5 micro-second twistor memory, of the "word selection" type, is presently under development at Bell Telephone Laboratories.

A third type of memory, "Persistor," based on the principle of superconductivity was discussed. Development of this memory element is being carried out at The Ramo-Wooldridge Corporation under the supervision of Dr. E. C. Crittendon, Jr. Low energy absorption, the prospect of greater combinatorial advantage, and the use of printed circuit assembly techniques show great promise in this type of memory.

• The August 1 meeting of the Mid Continent Computer (MC^2) Club in Chicago, Illinois, features a panel discussion of the operation of a computer installation. The moderator was Professor Robert Ashenhurst of the University of Chicago, and the panelists included Mr. Richard Haertle of A. C. Spark

Plug Division, General Motors, Professor Daniel Harris of Northwestern University, and Mr. Harry Hopp of Sinclair Research Labs.

- A "Computer Exhibition and Business Symposium," to be held in London, England, November 28–December 4, will be attended by more than forty British manufacturers of electronic computers and components. The symposium will stress the value of the computer as an aid to management and will present up-to-date information concerning installation and operation of computers and data processing systems and their applications to general business management, such as payrolls and stores control for railways, steel works, farms, and motor sales companies. Organizers are the National Research Development Corporation, the Electronic Engineering Association and the Office Appliance and Business Equipment Trades Association. Further information may be obtained by writing Exhibition Organizer, 11/13 Dowgate Hill, London, E. C. 4, England.

- The 1959 Electronics Components Conference, sponsored by the AIEE, IRE, EIA, and WCEMA, and scheduled to be held in Philadelphia, Pennsylvania, May 6–8, 1959, is soliciting technical papers in connection with its theme, "New Concepts for Space Age." Papers are requested on the following subjects: "Components and Their Application in Space Vehicles," "Micro-Miniaturization and The New Component—The Micro-Module," "Developments in Materials and Their Application to Components," "Professional and Tutorial Papers on The Mechanism of Failure (Operation) of Components," "New Techniques for Electronic Filtering, Tuning, and Switching," "Transistors, Diodes and Rectifiers," "Electron Tube Devices," "Instrumentation and Control Devices," "Survey of Improvements in Components," "New Developments in Passive Components," "Components for Military, Industrial and Home Appliance Applications of The Future," "Radiation Effects." Abstracts of 150 to 200 words in length together with title and author's name should be sent to the Technical Program Chairman: Brig. Gen. Edwin R. Petzing, AGEP Secretariat, University of Pennsylvania, 200 S. 33rd Street, Philadelphia 4, Pennsylvania, before October 1, 1958.

MIT LINCOLN LABORATORY MILESTONES*

J. T. GILMORE, JR.**

RETIRED. After 7 years of service the Memory Test Computer (MTC) was retired in February 1958 and is resting on its laurels. Achievements: test computer for first magnetic core memory; provided medium for circuitry and control-logic research; and terminal equipment research on the following: digital data transmitters and receivers; magnetic tape; microphone input; television input; Charactron tube displays. Applications: over 3000 hours of data reduction for the SAGE system; programming research in automatic coding, neuron network analysis, aircraft tracking studies, pattern recognition techniques, data processing, human engineering studies, weapons control simulation, and electronic counter-measures analysis and studies. Its greatest accomplishment, however, was the training of the people associated with it. The knowledge and experience they gained is a living memorial to a computer which became obsolete because of the research it helped make possible.

GRADUATED AND COLLEGE BOUND. The Lincoln Laboratory TX-O experimental computer; July 1958, after more than three years preparatory service as a testing ground for transistor circuitry and the 65,536-register ferrite-core memory. TX-O, to travel light, transferred the big memory to TX-2 and picked up a more compact 4096-register memory. It is now residing on the MIT campus where it will learn 8 more instructions and further increase its memory. In addition to the MIT Computation Center, Tech will now have an experimental computer for research data processing and new-equipment studies.

ADOPTED. By Lincoln Laboratory, June 1957; one IBM computer; Name: 704. Memory: 32,768 ferrite-core registers and 16,000 drum registers. Terminal equipment: card reader; card punch; line

*The research reported in this paper was supported jointly by the Army, Navy and Air Force under contract with the Massachusetts Institute of Technology.

**Staff Member, M.I.T. Lincoln Laboratory.

printer; 10 IBM magnetic tape units; cathode-ray tube display unit. Use: data reduction of SAGE ESS information; simulation of SAGE System; pattern recognition studies; computation center for Lincoln mathematicians.

BORN: One computer; Name: CG 24. First pulse: February 1958. Transistors: 18,000 (including memory drivers). Crystal Diodes: 33,000. Memory: 8192 ferrite-core registers (12 μsec cycle time). Addition Time: 24 μsec. Multiplication or division time: 84 μsec. Square root time: 300 μsec (to be reduced to 168 μsec late 1958). Word length: 25 bits plus two parity bits. Index Registers: 5. Instructions: 40. Terminal Equipment: direct input-output console typewriter; Photo electric reader; 2 cathode-ray tube display units with analog number generator; 3 real-time data input registers with direct access to core memory; 2 data output registers to position digital servomechanism; and 2 magnetic tape units to be added late 1958. Use: Real-time control and general scientific calculations. Location: Lincoln Laboratory Experimental Station, Millstone Hill, Westford, Massachusetts.

BORN: One special-purpose transistor computer; Name: ARC-1 (Average Response Computer). First pulse: March 1958. Transistors: 3000. Memory: 256 transistor driven ferrite-core registers (array fits in the palm of one's hand). Word length: 18 bits. Instructions: Wired program. Terminal Equipment: Datrac analog-to-digital converter; cathode-ray tube display unit; X-Y paper graph plotter; paper tape punch. Use: ARC-1 is being used at the MIT Biophysics Communications Laboratory to process experimental neurophysiological data (from animals) consisting of electrical brain responses evoked by repeated stimuli. The main job of the computer is to average the responses and display the resultant, thereby eliminating the background activity "noise" which is not "time locked" to the stimuli. It can also compile and display a frequency distribution of amplitudes at a specified time following the stimulus. Because of its 10 μsec cycle time ARC-1 is fast enough to compute and display results during the progress of the experiment.

BORN: One transistor computer; Name: TX-2. First Pulse: June 1958. Transistors: 30,000. Memory: 65,536 tube-driven ferrite-core registers (6 μsec), 4096 transistor-driven ferrite-core registers (5.5 μsec). Word length: 36 bits plus parity bit and meta bit.[1] Instruction cycle: 6 μsec. Addition (36 bits): 150,000 per sec., Multiplication (36 bits): 80,000 per sec., Addition and multiplication (9 bit words): 600,000 per sec. Instruction counters: 32 (Multiple Sequenced). Index registers: 32. Operand configuration registers: 9 wired, and 4 flip-flop (to be replaced by 32 thin-film registers by 1959). Instructions: 64. Terminal equipment: Photoelectric paper tape reader (1500 lines per sec); Soroban paper tape punch (240 lines per sec); Charactron Xerographic printer (2000 characters per sec); 2 direct input-output console typewriters; 16 magnetic tape units; 2 cathode-ray tube display units; 2 photo-diode light pens; 1 Epsco Datrac analog-to-digital converter. Use: A proving ground for new techniques, and a research tool for applications which cannot be easily done by the other computers in the laboratory.

[1] A meta bit is not changed by normal data transfers but only by an instruction that senses and/or sets individual bits of a given word. The meta bit is used to tag words or instructions which are of special interest to the programmer i.e., error diagnosis, etc.

COMING EVENTS

- International Association for Analog Computation—Second International Conference
  September 1–7, 1958; University Palace, Strasbourg, France
- Second International Congress on Cybernetics
  September 3–10, 1958; Namur, Belgium
  Contact: Secretariat, International Congress on Cybernetics, 13 rue Basse, Marcelle, Namur, Belgium
- SHARE Meeting
  September 10–12, 1958; St. Francis Hotel, San Francisco, California
- 13th Annual Instrument-Automation Conference and Exhibit (International)
  September 15–19, 1958; Philadelphia Convention Hall, Philadelphia, Pa.
  Sponsor: ISA
  Contact: J. F. Tabery, 3443 S. Hill Street, Los Angeles 7, California

- Meeting on Hydrodynamics
  October 16–18, 1958; Los Alamos Scientific Labs, Los Alamos, New Mexico
  Sponsor: Los Alamos Scientific Labs
  Contact: Dr. G. N. White, Los Alamos Scientific Labs
- UNIVAC Users' Conference
  October 20–21, 1958; John Hancock Mutual Life Insurance Company, Boston, Massachusetts
- National Business Show
  October 20–24, 1958; Coliseum, New York City, New York
  Contact: R. Lang, Managing Director, 530 Fifth Avenue, New York 36
- 1958 National Simulation Conference
  October 23–25, 1958; Statler-Hilton Hotel, Dallas, Texas
  Sponsors: IRE-PGEC and Dallas Section of IRE
  Contact: J. E. Howard, 2100 Menefee Drive, Arlington, Texas
- American Mathematical Society Meeting
  October 25, 1958; Princeton University, Princeton, New Jersey
- Fifth Annual Computer Applications Symposium
  October 29–30, 1958; Morrison Hotel, Chicago, Illinois
  Sponsor: Armour Research Foundation, Illinois Institute of Technology
  Contact: Armour Research Foundation, 35 West 33 Street, Technology Center, Chicago 16, Illinois
- 4th Electronic Business Systems Conference
  October 30–31, 1958; Olympic Hotel, Seattle, Washington
  Sponsor: Western Division, NMAA
  Contact: E.B.S. Conference, NMAA, Western Division, P. O. Box 134, Seattle 11, Washington
- International Conference on Scientific Information
  November 16–21, 1958; Mayflower Hotel, Washington, D. C.
  Sponsor: National Academy of Sciences, National Research Council, National Science Foundation, American Documentation Institute
  Contact: Secretariat, International Conference on Scientific Information, National Academy of Sciences, 2101 Constitution Avenue, N. W., Washington 25, D. C.
- American Mathematical Society Meetings
  November 21–22, 1958; Pomona, California
  and
  November 28–29, 1958; Northwestern University, Evanston, Illinois
  and
  November 28–29, 1958; Durham, North Carolina
- Computer Exhibition and Business Symposium
  November 28–December 4, 1958; London, England
  Contact: Exhibition Organizer, 11/13 Dowgate Hill, London E.C. 4, England
- Eastern Joint Computer Conference
  December 3–5, 1958; Bellevue-Stratford Hotel, Philadelphia, Pennsylvania
  Contact: John Broomall, Dev. Engr., Burroughs Corporation, Research Center, Paoli, Pennsylvania (General Chairman); F. M. Verzuh, Asst. Director, Computation Center, M.I.T., Cambridge, Massachusetts (Technical Program Chairman)
- American Mathematical Society—65th Annual Meeting
  January 20–22, 1959; University of Pennsylvania, Philadelphia, Pennsylvania
- Western Joint Computer Conference
  March 3–5, 1959; Fairmont Hotel, San Francisco, California
  Contact: M. L. Lesser, IBM Research Lab, San Jose, California
- Joint Meeting of Institute of Mathematical Statistics (Central Region) and the Association for Computing Machinery
  April 2–4, 1959; Case Institute of Technology, Cleveland, Ohio

Contact for IMS: Martin B. Wilk, Bell Telephone Laboratories, Murray Hill, New Jersey
Contact for ACM: Daniel Teichroew, National Cash Register, Dayton 9, Ohio

- 1959 Electronic Components Conference, "New Concepts for Space Age"
  May 6–8, 1959; Benjamin Franklin Hotel, Chestnut Street at 9th, Philadelphia, Pennsylvania
  Sponsors: AIEE, IRE, EIA, WCEMA
- First International Conference on Information Processing (ICIP)
  June 13–21, 1959; Europe
  Contact for U. S. Committee of ICIP: I. L. Auerbach, Auerbach Electronics Corporation, 109 North Essex, Narberth, Pennsylvania
  Sponsor: UNESCO
- 1959 ACM National Conference
  Summer, 1959; Massachusetts Institute of Technology, Cambridge, Mass.
  Contact: F. Verzuh, M.I.T.
- American Mathematical Society Meeting
  Summer, 1959; Salt Lake City, Utah
  November, 1959: Detroit, Michigan
- Eastern Joint Computer Conference
  1959; Boston, Massachusetts
- Second Interkama—International Congress and Exhibition for Measuring Techniques and Automation
  October 19–26, 1960

NOTICES

OF CHANGE OF ADDRESS

should be sent to the

ASSOCIATION FOR COMPUTING MACHINERY

2 East Sixty-Third Street
New York 21, New York.

●

One month's notice is required in order to guarantee delivery of periodicals and other mailings to the new address.

"We taught ourselves to use the Bendix G-15 computer ...it's that easy."

PETER M. LANG, *Senior Engineer*
NUCLEAR PRODUCTS—ERCO DIVISION OF ACF INDUSTRIES, INC.

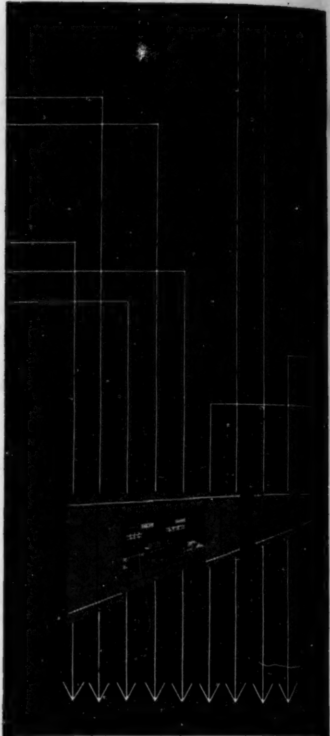
"Our nuclear engineering staff, in developing commercial power reactors, required a full-sized digital computer that the entire group could use. We picked the G-15 and I was first to learn its use. I taught myself in two days without help and in turn taught two-day classes for other ACF employees. Many of us had no previous computer experience, but we are now keeping our G-15 'hopping'—often seven days a week and eight to twelve hours a day."

THE G-15 GIVES YOU memory and speed of computers costing four times as much • Typewriter input-output, paper tape output and 250 characters/sec paper tape input at no added cost • Punch card input-output available • Extensive library of programs furnished • Strong user's sharing organization • Proven reliability • Nationwide sales and service • Lease or purchase.

Bendix Computer

DIVISION OF BENDIX AVIATION CORPORATION

Built and backed by Bendix, the G-15 is serving scores of progressive businesses large and small throughout the world. For details, write to Bendix Computer, Department R-4, Los Angeles 45, California.



INFORMATION SEARCHING and RETRIEVAL

A new and major program at The Ramo-Wooldridge Corporation is devoted to the design and development of a large-scale system for the automatic handling of reconnaissance information. The basic systems problems include the handling of ordinary language on computers and the design of automatic searching and retrieval techniques.

Inquiries are invited from electrical engineers, mathematicians and physicists whose backgrounds include operations research analysis and systems analysis of digital computing equipment.

For additional information, write to **Mr. Leslie Levin.**

The Ramo-Wooldridge Corporation

P. O. BOX 45215, AIRPORT STATION
LOS ANGELES 45, CALIFORNIA

then re

Electro

tape re

a new

simplif

trol sy

width

700 fe

and dr

reader

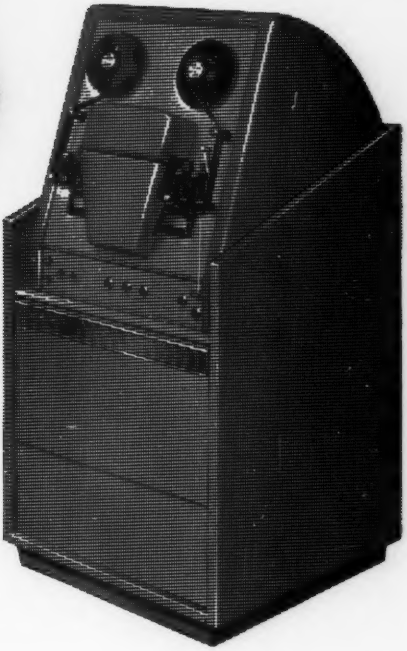
alread

details

On dis

Burroughs Photoreader

reads this many characters in one second!



1,000 CHARACTERS PER SECOND! And it stops on a single character... then reads the next character within five milliseconds after restart! The new Burroughs Photoreader by ElectroData Division of Burroughs Corporation is now commercially available... the finest precision paper tape reader of such high-speed performance being offered as a component. Speed of the Photoreader introduces a new high in computer-time efficiency to business and scientific data processing. Its instantaneous stop-ability simplifies computer techniques... also brings faster, more efficient operation to missile test checkout, fire control systems, equipment test procedures and machine automation. The Photoreader is adaptable to standard-width tape, from five to eight level code. Economic plastic reels are available in two sizes for tapes of 350 or 700 feet (40,000 or 80,000 Characters). Automatic rewind and end-of-tape sensing; true straight-line loading and drift-free design. Developed as an input unit for the new Burroughs 220 data processing system, the Photoreader is also available as a component for mounting in any standard 19″ cabinetry. It may also be ordered already housed in the Burroughs 220 cabinet, as pictured. For complete details write

Burroughs Corporation
ELECTRODATA DIVISION
PASADENA, CALIFORNIA

stops on a single character!

On display ISA Show – September 15-19 – Booth #1045

EXPLORE A NEW, UNCHARTED AREA IN INFORMATION HANDLING At—RCA's NEW

Astro-Electronic Products Division, Princeton, New Jersey

This new division makes available immediately creative opportunities in an advanced and challenging area of automatic data processing. The long-range research program now beginning will involve the solution of logic problems and the programming of deductive rules (non–numeric) and systems engineering.

Technical proficiencies are needed in all branches of information handling including the following areas of specialization: information systems design, development, analysis and synthesis; programming research and logical design; automatic language-translation and information retrieval; and digital computer design and engineering.

To participate in this program applicants must have advanced technical degrees, or equivalent experience, and a record of achievement. A government security clearance is required of all project personnel.

These opportunities offer stimulating positions in an atmosphere conducive to creative activity as part of the Princeton research community. For a personal interview, communicate with Mr. D. D. Brodhead.

RCA®

RADIO CORPORATION
OF AMERICA
Astro-Electronic Products Division
Princeton, New Jersey

Call Collect, WAlnut 4-2700
Or Send Resume to Dept. PE-16

EQUAL JOB
For All Qualified Applicants
OPPORTUNITY

ital

ust
ent
ern-
ect

s in
part
nal
l.